CITY UNIVERSITY OF HONG KONG
香港城市大學

CROSS-MODAL COOKING RECIPE RETRIEVAL
跨媒體菜譜檢索

Submitted to
Department of Computer Science
電腦科學系
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
哲學博士學位

by

Chen Jingjing
陳靜靜

June 2018
二零一八年六月

ABSTRACT

This thesis investigates the problem of cross-modal cooking recipe retrieval from four aspects: (1) recognizing ingredients in food images and building ingredient graph for zero-shot recipe retrieval, (2) recognizing rich attributes of food, not only ingredient but also cooking, cutting methods (3) learning joint embedding space between ingredients (extracted from recipe) and food images with attention modeling, and (4) deep understanding the cooking instructions for cross-modal learning.

We first focus on the recognition of ingredients for recipe retrieval in the domain of Chinese dishes. Different from food categorization, which is to identify the name of a dish, ingredient recognition is to uncover the ingredients inside a dish. As the size, shape and color of ingredients can exhibit large visual differences due to diverse ways of cutting and cooking, in addition to changes in viewpoints and lighting conditions, recognizing ingredient is much more challenging than food categorization. We propose deep architectures for simultaneous learning of ingredient recognition and food categorization, by exploiting the mutual but also fuzzy relationship between them. The learnt deep features and semantic labels of ingredients are then innovatively applied for zero-shot retrieval of recipes. Besides, to boost retrieval performance, a graph encoding the contextual relationship among ingredients is learnt from the recipe corpus. Using this graph, conditional random field (CRF) is employed to probabilistically tune the probability distribution of ingredients to reduce potential recognition error due to unseen food category.

As similar ingredient composition can end up with wildly different dishes depending on the cooking and cutting procedures, the difficulty of retrieval originates from fine-grained recognition of rich attributes from pictures. We therefore proposed multi-task learning to learn not only the ingredient composition but also

the applied cooking and cutting methods. The proposed model suffers less from the need of a large amount of learning samples and is easier to be trained with a smaller number of network parameters. With a multi-task deep learning model, we provide insights on the feasibility of predicting ingredient, cutting and cooking attributes for food recognition and recipe retrieval. Besides, as the learning happens at region-level, localizing the ingredient is also possible even when region-level training examples are not provided.

Training deep models for ingredient recognition requires manually labeling the ingredient, which is expensive and time-consuming. As there are already millions of food-recipe pairs that can be acquired from the Internet, a more feasible means that can save labeling efforts is to learn the joint space between recipes and food images for cross-modal retrieval. Therefore, we exploit and revise a deep model, stacked attention network for joint embedding feature learning between dish images and ingredients extracted from cooking recipes. Given a large number of image and recipe pairs acquired from the Internet, a joint space is learnt to locally capture the ingredient correspondence from images and recipes. As learning happens at the region level for image and ingredient level for recipe, the model has the ability to generalize recognition to unseen food categories.

To further improve the overall retrieval performance, we explore utilizing cooking instruction for cross-modal learning. Cooking instruction, on the one hand, gives clues to the multimedia presentation of a dish (e.g., taste, color, shape). On the other hand, describes the process implicitly, implying only the cause of dish presentation rather than the visual effect that can be vividly observed on a picture. Therefore, different from other cross-modal retrieval problems in the literature, recipe search requires the understanding of textually described procedure to predict its possible consequence on visual appearance. We approach this problem from the perspective of attention modeling. Specifically, we model the attention

of words and sentences in a recipe and align them with its image feature such that both text and visual features share high similarity in multi-dimensional space. Furthermore, with attention modeling, we show that language-specific named-entity extraction based on domain knowledge becomes optional.

The proposed techniques are evaluated on large-scale real-world food image and recipe dataset including VireoFood 172, UEC-Food100 and recipe1M. Experimental evaluations demonstrate promising results of our techniques and show good potential for real-world multimedia applications.

CITY UNIVERSITY OF HONG KONG
Qualifying Panel and Examination Panel

Surname: CHEN

First Name: Jingjing

Degree: PhD

College/Department: Department of Computer Science

The Qualifying Panel of the above student is composed of:

Supervisor(s)
Prof. NGO Chong Wah — Department of Computer Science, City University of Hong Kong

Qualifying Panel Member(s)
Dr. CHAN Antoni Bert — Department of Computer Science, City University of Hong Kong

Dr. WONG Hau San — Department of Computer Science, City University of Hong Kong

This thesis has been examined and approved by the following examiners:

Prof. LI Qing — Department of Computer Science, City University of Hong Kong

Prof. NGO Chong Wah — Department of Computer Science, City University of Hong Kong

Dr. CHEUNG Kwok Wai William — Department of Computer Science, Hong Kong Baptist University

Dr. IDE Ichiro — Graduate School of Informatics, Nagoya University

ACKNOWLEDGEMENTS

I would never finish my dissertation without the help from numerous talent people. Deep thanks to everyone that supports me in the whole study.

First of all, I would like to express my sincere gratitude to my advisor Prof. Chong-Wah Ngo for the continuous support of my Ph.D study, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. His rigorous working style and logical thinking value much beyond this Ph.D. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to thank Prof. Tat-Seng Chua, my advisor at National University of Singapore, for his continuous supports during my fourteen months exchanging period in LMS group. His critical think and valuable advices greatly broadened my scope of research.

Also I would like to thank my examiners, Prof. Qing Li, Prof. Ichiro Ide and Prof William Cheung, for their precious time and valuable feedbacks. I am also grateful to my qualifying panel members, Dr. Hau San Wong and Dr. Antoni Bert Chan, for their constructive suggestions and comments on my research reports.

Furthermore, I would like to express my thanks to former and current colleagues in VIREO group at the City University of Hong Kong for their support, inspiration, enjoyable time we have spent together - Xiao Wu, Xiao-Yong Wei, Yu-Gang Jiang, Wan-Lei Zhao, Shi-Ai Zhu, Song Tan, Ting Yao, Zhi-Neng Chen, Wei Zhang, Chun-Chet Tan, Lei Pang, Hao Zhang, Yi-Jie Lu, Maaike de Boer, Zhao-Fan Qiu, Qing Li, Phuong-Anh Nguyen and Thanh Nguyen Huu, Yanbin Hao, Bin Zhu. Also I want to thank the colleagues in LMS lab at National University of Singapore for their supports in every way: Jing-Yuan Chen, Na Zhao, Di Xin, Xiao-Yu Du, Yan-Zhao Ming, Li-Zi Liao, Yun-Shan Ma, Fu-li Feng, Xiang-Wang,

Xiang-Nan He, Francesco Gelli.

Last, I want to express my deepest gratitude to my parents, for their everlasting love and support, my younger sister, for significant positive effect. Also I would like to thank Jing-Wei Hu, my boyfriend. Understanding me best as a Ph.D. himself, Jing-Wei has been a great companion, loved, encouraged, entertained, and helped me during the past years in the most positive way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Cross Modal Recipe Retrieval: Motivation and Challenges

Food intake tracking has recently captured numerous research attentions [3] [4] [5] for long-term impact of food consumption on health. The main pipeline of tracking is to take a picture of the dish, recognize its category and then search for relevant sources for nutrition and calories estimation [6] [7]. The sources are usually food labels and food composition tables (FCT) compiled by nutrition experts [8]. Nevertheless, in the free-living environment, dishes are often prepared in wild with no expert references for health index estimation.

The prevalence of sharing food images and recipes on the Internet [9], nevertheless, provides a new look to this problem. Specifically, there are social media platforms in both eastern and western countries, such as "Go Cooking" [1] and "All Recipes" [2], for master and amateur chefs to share their newly created recipes and food images. There are also followers or fans that follow the cooking instructions in recipes to reproduce the same dishes and upload their pictures to websites for peer comment. To date, these websites have accumulated over millions of recipes and images. These recipes are mostly listed with ingredients alongside with their quantities, supplying a new source of references for food intake tracking. Furthermore, cooking procedure, i.e., how ingredients are prepared and cooked (e.g., deep

[1] https://www.xiachufang.com/

[2] https://www.allrecipes.com/

fried versus steam), provides another dimension of clues which is not listed in food label or FCT for health management. Hence, in principle, being able to link a food image to its right recipe available on the Internet will facilitate the evaluation of nutrition contents.

In this thesis, we focus on the problem of retrieving recipes for given food images. Retrieving recipes corresponding to the given food images pictures is a difficult problem. The challenge mainly comes from three aspects. First, understanding the contents of the food image (i.e., category, ingredient composition, cooking and cutting methods) is challenging. As shown in Figure 1.1, food images have large visual variations in terms of shape, color, and texture layout even within a food category, which makes food recognition much more difficult than other object recognition. Besides, automatic recognition is also challenged by the wildly different ways of mixing and placing ingredients even for the same food category. For the food category "steamed egg custard" (last row of Figure 1.1), there is even no overlap in ingredients except for egg. Secondly, as recipe contains richer information (i.e., food name, ingredients, quantity, taste, cooking procedure), how to properly model and represent recipe remains a challenge. Besides, online recipes are written in free form with user-generated text and are difficult to be syntactically or semantically analyzed. Lastly, the recipe contains ingredients or cooking procedures that are not directly translatable to image content. For example, invisible ingredients such as "salt", "honey" or instructions that have no effects on the dish appearance such as "washing the cucumber" and "preheat the oven". As results, how to build the correspondence between food image and recipe is also challenging.

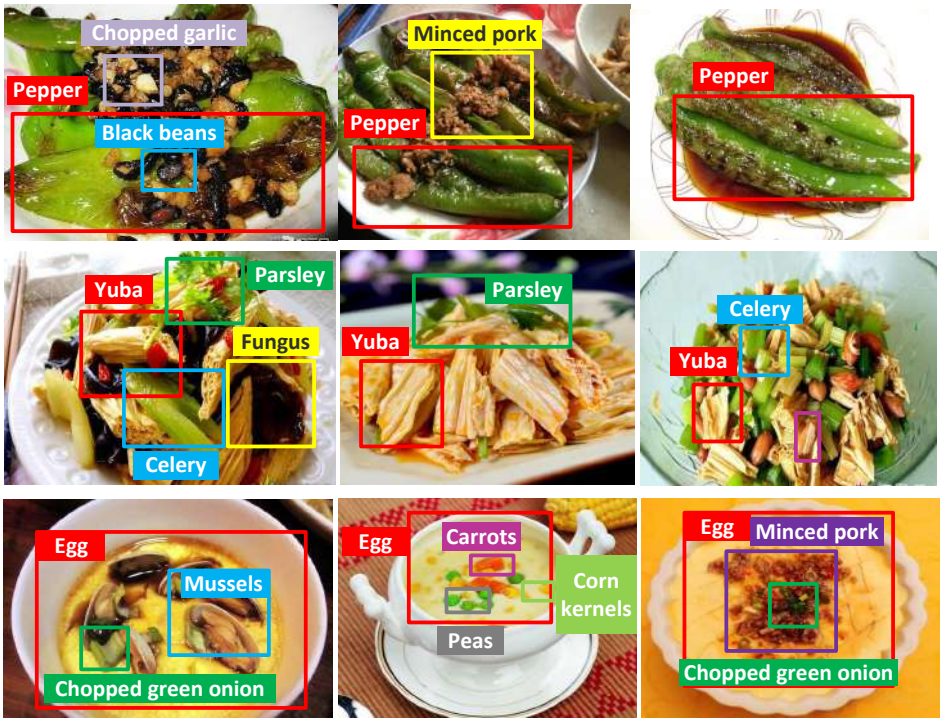


Figure 1.1: Variations in visual appearance and composition of ingredients show the challenges of predicting ingredients even for dishes within the same food category. The first row shows three examples of dishes for the category "fried green peppers", followed by "yuba (a food product made from soybeans) salad" and "steam egg custard" in the second and third rows respectively.

1.2 Thesis Overview and Contributions

This thesis investigates cross-model cooking recipe retrieval. Specifically, to address the aforementioned challenges, we study this problem from two perspectives: recognition based recipe retrieval and cross-modal learning based retrieval. For the former, we study ingredients recognition and rich attribute learning (ingredients, cooking and cutting methods) while for the later, we study stacked attention modeling on food image and deep understanding of cooking recipes with hierarchical attention model. In the following, we overview the proposed methods and brief the challenges and contributions.

1.2.1 Ingredient recognition

In general, recognition of dish ingredients, such as "yuba" (a food product made from soybeans) and "green onion" in Figure 1.1, could be even harder than food categorization. In addition to variations because of cutting and cooking methods, the scale, shape and color of ingredients exhibit wild differences under various viewpoints, lighting conditions and occlusions. This thesis considers simultaneous recognition of food and ingredients, hoping to exploit the mutual relationship between them for boosting respective recognition performances. Specifically, the key ingredients (e.g., egg in "steamed egg custard") within a category remain similar despite variation in auxiliary ingredients (e.g., mussel, minced pork). Treating key ingredients as "attributes" ideally could assist in food categorization. On the other hand, if food category is known, the prediction of ingredients could also become easier. For example, the chance that "fungus" appears in "yuba salad" is much higher than "garlic". Hence, learning food categories with the composition of ingredients as knowledge, and vice versa, in principle shall lead to better performance than treating them as two separate recognition tasks.

Therefore, we formulated ingredient recognition as a problem of multi-task learning and explored different multi-task architectures for simultaneous learning of ingredient recognition and food categorization. For experimental purpose, a large Chinese food dataset VIREO Food-172, which is manually labeled with ingredient labels, has been constructed. The proposed multi-task deep models were evaluated on the VIREO Food-172 dataset, and the experimental results show that, compared to single-task model, multi-task model which introduces category information as supervision signals achieves better ingredient recognition performance. As the proposed multi-task model is capable of predicting ingredients, in principle the recipes of images from an unknown food category can be retrieved through

matching of ingredients. To boost retrieval performance, a graph encoding the contextual relationship among ingredients is learnt from the recipe corpus. Using this graph, conditional random field (CRF) is employed to probabilistically tune the probability distribution of ingredients to reduce potential recognition error due to unseen food category. With the aid of external knowledge, the recognized ingredients of a given food picture are matched against a large recipe corpus, for finding appropriate recipes to extract nutrition information. The recipe retrieval performances on unseen food categories demonstrate the feasibility of the proposed approach for zero-shot cooking recipe retrieval.

Contribution: Our main contribution is the proposal of multi-task learning model for ingredient recognition (Chapter 3.1) and demonstrates its application for zero-shot recipe retrieval (Chapter 3.2). Our work differs from the existing works, which mostly focus on recognition of food categories and operate in domains such as western and Japanese food [10] [11]. To our knowledge, zero-shot recipe retrieval, which requires knowledge of ingredients, has not yet been considered in the literature. In addition, we also release the collected Chinese food dataset, VIREO Food-172, which contains 172 food and 353 ingredient labels. The dataset is larger than the publicly available datasets such as Food-101 [10], UEC Food-100 [11] and PFID [12], each with around 100 western or Japanese food categories but without ingredient labels.

1.2.2 Rich Attribute Learning

As similar ingredient composition can end up with wildly different dishes depending on the cooking and cutting procedures, the difficulty of recipe retrieval originates from fine-grained recognition of rich attributes from pictures. Therefore, we also study the problem of rich food attribute recognition, not only the ingredient

composition but also the applied cooking and cutting methods. Specifically, the cooking and cutting attributes in our work are assigned locally with ingredients. Although associating cooking attributes globally with dishes as in [13] simplifies the design of deep architecture, the model cannot be employed for retrieving recipes where ingredients are individually cooked before composing into dishes.

Intuitively, having the cutting and cooking information enables better inference on the ingredients appearances, which will benefit ingredient recognition. However, a particular challenge is that the prediction of cutting and cooking attributes requires the knowledge of ingredient locations. In other words, ingredient regions have to be localized and recognition of attributes should happen at the image region level. For prepared foods where ingredients are mixed and stirred or scrambled, collecting region-level ingredient labels is extremely difficult. Therefore, a more feasible way is leveraging semi-supervised or unsupervised models proposed for semantic segmentation for ingredient localization.

To address the aforementioned challenges, we consider a multi-task learning model that performs the recognition of the ingredient, cooking and cutting methods at region level. Given a picture of dish, the multi-task model outputs the probability distributions of the ingredient, cooking and cutting methods at each region. Then, a new pooling technique, namely, dependency pooling is tailored to combine the results of region-level predictions to get image-level predictions for different tasks. As the prediction happens at region-level, the proposed model is able to localize ingredients even when region-level labels are not provided. The developed rich attribute learning model is able to recognize 1,276 ingredients, 10 cutting methods and 37 cooking methods. To the best of our knowledge, this is the first attempt to recognize cutting and cooking methods applied on ingredients. Besides, this is also the first work that considers the recognition for thousands of ingredients.

Contribution: The main contribution is on the introduction of rich food attributes for cross-modal recipe retrieval, which addresses the limitation of existing literature on how the ingredients are labeled and utilized for search. To the best of our knowledge, there is no research effort yet on the prediction and leveraging of all the three attributes for food recognition and recipe retrieval.

1.2.3 Cross-modal Learning with Stacked Attention Model

Compared with recognition-based recipe retrieval approaches, learning the joint space between recipe and food images for similarity ranking requires less labeling efforts, hence it is more scalable. Nevertheless, the major challenge of cross-modal learning in food domain is how to build the correspondence between recipes and food images. There could be many recipes named under the same category, each of which differs in the composition of ingredients. Figure 1.2 shows an example, where there are different version of "Yuba salad", and only one best match recipe that contains the same ingredients composition with the query image. The learnt joint space should be able to deal with such situation and retrieve the best match recipe. Therefore, in this thesis, we propose to build the correspondence between recipes and images based on the ingredients. Specifically, the correspondence is captured by stacked attention model on region level. The joint space is learnt between attended ingredient regions and ingredients extracted from recipes. As learning happens at the region level for image and ingredient level for recipe, the model has the ability to generalize recognition to unseen food categories.

Contribution: The main contribution is on the introduction of stacked attention model during the joint space learning between ingredients and food images. As the stacked attention mechanism has the ability to infer attended regions relevant to ingredients, the proposed model is able to achieve better best-match recipe

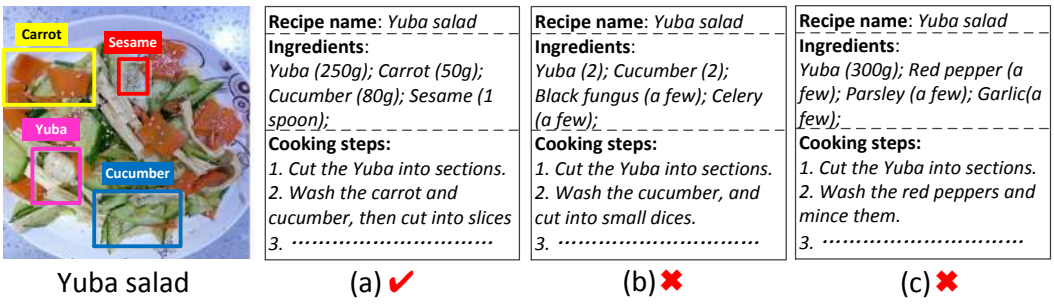


Figure 1.2: Although recipe (a), (b) and (c) are all about "Yuba salad", only recipe (a) uses the exactly same ingredients as the dish picture. Retrieving best-match recipe requires fine-grained analysis of ingredient composition.

retrieval performance even for unseen food categories.

1.2.4 Deep Understanding of Cooking Procedure

Cooking recipes contain rich information. As shown in 1.3, a recipe usually has three sections: title, ingredient and procedure. Title resembles phrase while ingredients can be regarded as keywords analog to the traditional visual annotation problem [14], which explicitly lists out the contents of a food image. Cooking instruction, on the other hand, is composed of a series of sentences detailing the food preparation and cooking process. In literature, leveraging cooking recipe for cross-modal analysis is either based on title or ingredient, as the cooking procedure is difficult to model. Online recipes are written in free form with user-generated text and are difficult to be syntactically or semantically analyzed.

Furthermore, different from problems such as image captioning [15] and visual question-answering [16], the descriptions in the cooking procedure are not directly translatable to image content. Rather, the instruction at a step dictates the causality of food preparation which may not be relevant to final food presentation or even be visible in food image. For example, the instructions "position rack about 4 inches from the boiler" in Figure 1.3(a) and "put in the garlic powder then

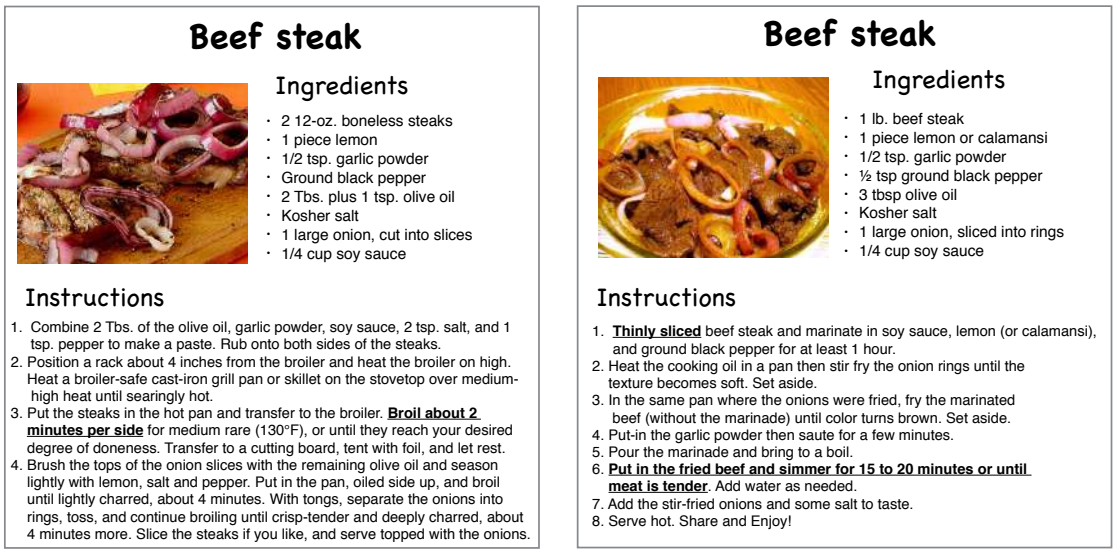


Figure 1.3: Understanding recipes is not easy even by human. Both dishes have the same name and almost similar ingredients, but are prepared in different manners and result in different presentations. The differences (e.g., broil versus simmer) are underlined to highlight the cause-and-effect in cooking procedure.

saute for a few minutes" in Figure 1.3(b) have insignificant outcome to the visual appearance of dishes. Furthermore, online recipes are user-generated and there are no rules governing the documentation of recipes. Sentences such as "Serve hot! Share and enjoy!" (Figure 1.3(b)) are visually irrelevant, "slice the steaks if you like" (Figure 1.3(a)) present visual uncertainty.

The purpose and format of recipe make the challenges of cross-modal retrieval different from other problem domains [14] [15] [16] in multimedia. As shown in Figure 1.3, both recipes have the same title and almost similar list of ingredients. However, the dish presentations exhibit different visual appearances beyond photometric changes due to differences in the cooking process. Precisely, the steak in Figure 1.3(a) is broiled while the steak in Figure 1.3(b) is fried and simmered. In addition, some ingredients are used in different stages for different purposes. For example, lemon in 1.3(a) is seasoned on onion slice, and lemon in Figure 1.3(b)

is mixed with other sauces to marinate the beef steak. These procedural descriptions do not directly link to the visual appearance but have an implicit impact on the final food presentation. Furthermore, the relationship of cooking and cutting actions to the visual appearance of food is not always one-to-one, but intertwines with types of ingredients and seasonings being added.

In this thesis, we propose a hierarchical attention mechanism based on [17] to model the complex word-to-word and sentence-to-sentence interactions in the recipe as a vector. Instead of modeling recipe as an action graph illustrating the flow of food preparation [18] [19] [20], embedding a recipe into a vector representation that captures word and sentence significances is more feasible with the rapid advancement of deep learning.

Contribution: The main contribution of this work is the embedding of a recipe into a vector representation for capturing the cooking procedure that implies the causality effect between ingredients and actions. The resulting vector is represented in a form similar to a visual vector, allowing parameter tuning and data-driven search of weights to align the relevancy of words or sentence to visual content.

1.3 Organization

The rest of this thesis is organized as follows. Chapter 2 reviews related works on food image and recipe analysis, including food categorization, ingredient recognition, food attribute recognition, cross-modal analysis and recipe analysis. Chapter 3 presents our approach for ingredient recognition and zero-shot recipe retrieval. In Chapter 4, we introduce our region-wise multi-task model for rich food attribute learning, and further leverage the learnt rich food attribute for cross-model retrieval. Chapter 5 presents a stacked attention model for cross-modal

learning of images and recipes. In Chapter 6 we present a hierarchical attention model for deep understanding of cooking procedure. Finally, Chapter 7 concludes the thesis and discusses our future research directions.

1.4 Publication

The work presented in Chapter 3 was published in ACM Multimedia 2016 [21]. The work discussed in Chapter 4 was published in ACM Multimedia 2017 [22]. The material introduced in Chapter 5 was published in Multimedia Modeling 2017 [23], and the extended version was published in Multimedia Tools and Applications [24]. The work in Chapter 6 is accepted by ACM Multimedia 2018.

In addition to the works presented in this thesis, I have also worked on several other problems during my Ph.D. study, including instance search [25], customized cooking assistant system [26] and dietary tracking system [7]. Interesting readers can refer to these publications for technical details.

CHAPTER 2

LITERATURE REVIEW

This chapter gives a literature overview of food-related tasks, including food recognition, cross-modal analysis and cooking recipe analysis. Comparisons and contrasts between the existing methods and the proposed works are also discussed.

2.1 Food recognition

Variants of recognition-centric approaches have been investigated for different food-related applications. These efforts include food quantity estimation based on depth images [27], image segmentation for volume estimation [28], context-based recognition by GPS and restaurant menus [29], taste estimation [30], multiple-food recognition [11], multi-modal fusion [20] and real-time recognition [31]. This section mainly reviews previous works in food categorization and ingredient recognition using hand-crafted and deep features. In addition, recent works on other food attributes recognition such as cooking methods, cuisine and course attributes are also discussed.

2.1.1 Food Categorization

The challenge of food categorization comes from visual variations in shape, color and texture layout. These variations are hard to be tackled by hand-crafted features such as SIFT [32], HOG [33] and color [34]. Instead, deep features extracted from DCNN [35], which is trained on ImageNet [36] and fine-tuned on food images,

often exhibit impressive recognition performance [37] [38] [39]. As studied in [40] [41], deeper networks such as VGG [42], GoogleNet [43] and Resnet [44] tend to generate better food features than AlexNet. In [41], a wide-slice residual network is proposed for food recognition and demonstrated to achieve better performance than AlexNet. In addition, combination of multi-modal features sometimes also leads to better recognition performance, as reported in [39] [45]. One of the best performances on UEC Food-100 dataset is achieved by fusion of DCNN features with RootHOG and color moment [45], and similarly for UPMC Food-101 dataset by fusion of textual and deep features [39].

Different from these works which directly adopt DCNN for food categorization, our research in Chapter 3 contributes by proposing new architectures based on DCNN for simultaneous recognition of food categories and ingredients. Since the proposal of multi-task in [21], multi-tasking learning in food domain have become popular [13] [46]. Reference [13] proposes to simultaneously recognize food category, ingredients and cooking methods while [46] proposes a multi-task model to simultaneously predict food categories and calories for food image. Reference [46] demonstrates that multi-task model could boost the performance of both food categorization and calories estimation.

2.1.2 Ingredient Recognition

Ingredient recognition receives much few attentions than food categorization [13] [47] [48]. The problem is more challenging as ingredients are small in size and can exhibit larger variances in appearance. Early studies include PFD (pairwise local feature distribution) [48], which defines 8 types of ingredients for pixel labeling. PFD explores spatial relationship between ingredient labels to generate high dimensional features for food recognition. Although powerful, PDF is not scalable as

feature dimension is exponential to the number of ingredients, and can grow dramatically to tens of thousands of dimensions with only 8 ingredients. In [10] [49], instead of explicitly defining ingredient labels for recognition, discriminative features corresponding to ingredients are mined for recognition. For example, DPM (deformable part-based model) and STF (semantic texton forest) are proposed in [49] for detection of ingredient regions. Random forest is employed in [10] to cluster super-pixels of food pictures for inferring prominent features for recognition. These approaches, while capable of showing excellent performance for standardized cooked food such as desserts and fast food, require tuning of hand-crafted parameters for performance optimization. More importantly, for dishes with wild composition of ingredients, learning of discriminative features is practically difficult to achieve with shallow methods such as feature clustering [10].

Deep-based ingredient recognition has also been recently investigated. In [13], a multi-task model learning framework is proposed for simultaneous recognition of food categories, ingredient labels and cooking attributes. In [47], multimodal deep belief network is used for ingredient recognition and food image retrieval. Another work is [50] which exploits the rich relationships among ingredient, food category and restaurant through bipartite-graph representation. Segmentation of food into ingredients is also explored in [37], by convolutional network and CRF (conditional random field). Nevertheless, this approach requires location labels for training, which are difficult or even impossible to be obtained for prepared foods with ingredients being cut and mixed or stirred.

In Chapter 3, we focus on exploring different multi-task models for ingredient recognition in Chinese food domain. Besides, an ingredient graph is built to encode the relations among ingredients for improving the ingredient recognition performance and zero-shot recipe retrieval. Our work differs from the existing works which mostly focus on recognition of food categories and operate in domains such

as western and Japanese food [10] [11]. To our knowledge, zero-shot recipe retrieval, which requires knowledge of ingredients, has not yet been considered in the literature.

2.1.3 Food Attribute Recognition

Food is rich in visible (e.g., ingredients) and procedure (e.g., cooking, cutting) attributes. Apart from ingredients, other attributes such as venue, GPS, cuisine (e.g., Chinese, American), course (e.g., breakfast, afternoon tea) and cooking methods, have also been exploited in recent years [29] [13] [47] [51]. Basically, most of these works adopt multi-task learning for food attributes recognition. In [47], a multi-modal multi-task deep belief network is proposed for cuisine and course classification. Reference [13] proposes a multi-task learning model for simultaneous recognition of food categories, ingredient labels and cooking attributes. However, in this approach, cooking attributes are assumed to be globally associated with dishes and not locally with ingredients. Although this assumption simplifies the design of deep architecture, the model cannot be employed for retrieving recipes where ingredients are individually cooked before composed into dishes. Different to the aforementioned works, our work presented in Chapter 5 aims to predict cooking and cutting attributes at ingredient level. In our proposal, the cooking and cutting attribute can be predicted without the location information of ingredients. As far as we known, this is the first work that investigates the interplay between visual and procedural attributes for cross-modal recipe retrieval.

2.2 Cross-modal analysis

Cross-modality analysis has been actively researched for multimedia retrieval [52] [53] [54]. Frequently employed algorithms include canonical correlation analysis

(CCA) [55] and partial least squares (PLS) [56], which find a pair of linear transformation to maximize the correlation between data from two modalities. CCA, in particular, has been extended to three-view CCA [57], semantic correlation matching (SCM) [54], deep CCA [58] and end-to-end deep CCA [59] for cross-modality analysis. Recent approaches mostly rely on deep learning, for examples, deep CCA [59], DeViSE [52], correspondence auto-encoder [60] and adversarial cross-modal retrieval [61]. These models, nevertheless, consider image-level features, such as fc7 extracted from deep convolutional network (DCNN), and usually ignore regional features critical for fine-grained recognition. One of the exceptions is the deep fragment embedding (DFE) proposed in [53], which aligns image objects and sentence fragments while learning the visual-text joint feature. However, the model is not applicable here for requiring of R-CNN [62] for object region detection. In the food domain, there is yet to have an algorithm for robust segmentation of ingredients, which can be fed into DFE for learning.

Cross-modal learning in food domain has started to attract research interest in recent years, and several large food and recipe datasets have been developed recently, for example, Cookpad [63] and Recipe1M [2] datasets. Existing efforts include [2] [39] and [47]. In [47], deep belief network is used to learn the joint space between food images and ingredients extracted from recipes. This approach considers image-level features for joint-space learning. Different from [47], our work presented in Chapter 4 aims to learn a joint space between food images and ingredients on regional level with a stacked attention network. The stacked attention is able to simultaneously locate ingredient regions in an image and learn multimodal embedding features. Reference [2] also studies learning the joint embedding space between food images and recipes. Different from reference [47] where the recipes are represented by attributes (i.e., ingredient), the recipe representation in [2] is learnt by encoding ingredients and cooking instructions using recurrent neu-

ral networks. Our work presented in Chapter 6 differs from [2] for incorporation of word-level and sentence-level attentions at three different levels of granularity (i.e., title, ingredient, instruction) for representation learning.

2.3 Recipe analysis

Analysis of recipes has been studied from different perspectives, including retrieval [2] [21] [39] [64], classification [30] [65] and recommendation [66]. Most of the approaches employ text-based analysis based upon information extracted from recipes. Examples include extraction of ingredients as features for cuisine classification [65] and taste estimation [30]. More sophisticated approaches model recipes as cooking graphs [64] [67] such that graph-based matching can be employed for similarity ranking of recipes. The graph, either manually or semi-automatically constructed from a recipe, represents the workflow for cooking and cutting procedures of ingredients. In reference [64], multi-modal information is explored, by late fusion of cooking graphs and low-level features extracted from food pictures, for example-based recipe retrieval. Few works have also studied cross-modality retrieval [39] [66]. In [66], recognition of raw ingredients is studied for cooking recipe recommendation. Compared to prepared food where ingredients are mixed or even occlude each other, raw ingredients are easier to recognize. In reference [39], a classifier-based approach is adopted for visual-to-text retrieval. Specifically, the category of food picture is first recognized, followed by retrieval of recipes under a category. As classifiers are trained from UPMC Food-101 dataset [10], retrieval is only limited to 101 food categories. The issues in scalability and finding best-match recipes are not addressed. Different from reference [39], the work presented in this thesis retrieves recipes in a more scalable way. By recognizing ingredients in food images and matching against the ingredient list extract from

text recipes, our work is able to address the best-match recipe retrieval issue. By learning the joint space between food images and text recipes, our work is demonstrated to have higher generalization ability and can even retrieve recipes for food from unknown categories.

CHAPTER 3

INGREDIENT RECOGTION

In the literature, associating food categories to their respective recipes is regarded as a general pipeline that facilitates the estimation of calories and nutrition facts [68] [31]. The pipeline is effective for recognizing restaurant dishes and the food categories with standardized cooking method (e.g., fast food) that often have similar visual appearance with the same ingredients. However, most dishes in Chinese food have no standardized cooking method, food presentation and ingredient composition. Direct mapping between dishes and recipes, by using the names of food categories, is not likely to attain satisfactory retrieval rate, not mentioning the imperfect performance in food recognition. The difficulty of this task is probably alleviated, nevertheless, with the presence of GPS and restaurant menus as utilized by Im2Calories [37] and Menu-Match [69]. However, restaurant information is difficult to acquire as stated in [37] and such context-aware recognition is only limited to restaurant food. Therefore, this thesis argues the need of ingredient recognition beyond food categorization for general recipe retrieval. As the number of food categories is generally far larger than the number of ingredients, recognizing attributes is more feasible than food categories in terms of scale. Furthermore, ingredient recognition also gives light to the retrieval of recipes for unknown food categories during model training, a problem generally referred to as zero-shot recognition or retrieval [70].

This Chapter studies the recognition of ingredients for recipe retrieval in the domain of Chinese dishes. Different from food categorization, which is to identify the name of a dish, ingredient recognition is to uncover the ingredients inside a dish

(e.g., green pepper, black bean, chopped garlic). Generally speaking, ingredient recognition is more difficult than food categorization. The size, shape and color of ingredients can exhibit large visual differences due to diverse ways of cutting and cooking, in addition to changes in viewpoints and lighting conditions. Recognizing ingredients alone without food category in mind is likely to result in unsatisfactory performance. This Chapter considers simultaneous recognition of food and ingredients, aiming to exploit the mutual relationship between them for enhancing the robustness of recognition. The key ingredients of a category remain similar despite composing with different auxiliary ingredients. Knowing food category basically eases the recognition of ingredients. On the other hand, the prediction of ingredients also helps food categorization, for example, the ingredient "fungus" has a higher chance than "pork" to appear in the food "yuba salad". Hence, learning food categories with the composition of ingredients in mind, and vice versa, in principle shall lead to better performance.

Figure 3.1 gives an overview of the proposed framework, which is composed of two modules: ingredient recognition and zero-shot recipe retrieval. The first module formulates the recognition of ingredients as a problem of multi-task learning using deep convolution neural network (DCNN). Given a picture of a dish, the module outputs the name of the dish along with a histogram of ingredients. The developed DCNN can recognize 172 Chinese food categories and 353 ingredients. To the best of our knowledge, there is no result published yet for ingredient recognition on such a large scale. The second module performs zero-shot retrieval, by matching the predicted ingredients against a large corpus containing more than 60,000 recipes. The corpus includes some food categories as well as ingredients unknown to the multi-task DCNN. To boost retrieval performance, a graph encoding the contextual relationship among ingredients is learnt from the recipe corpus. Using this graph, conditional random field (CRF) is employed to probabilistically

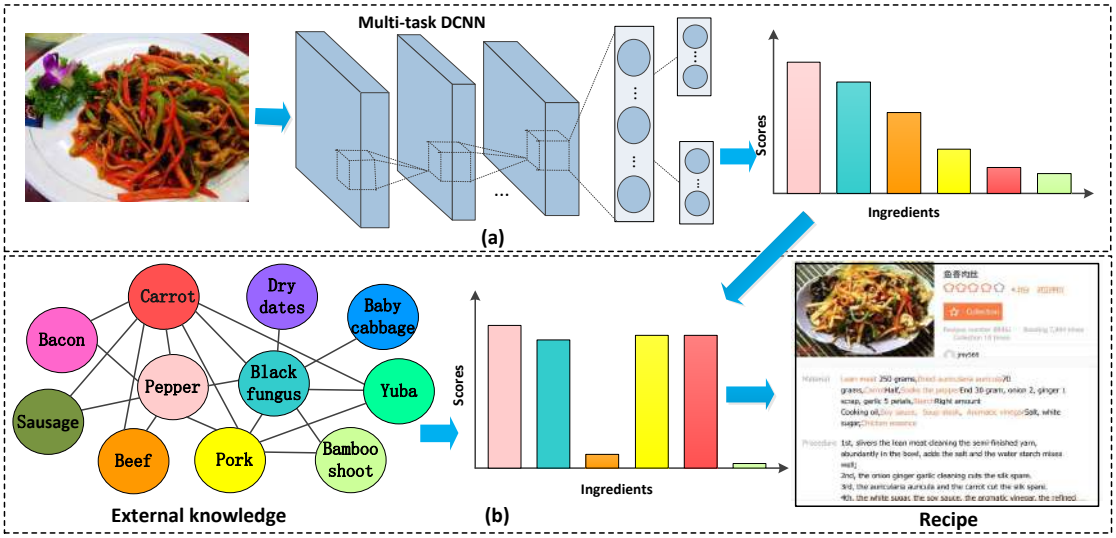


Figure 3.1: Framework overview: (a) ingredient recognition, (b) zero-shot recipe retrieval. Given a picture of dish with unknown food category, the framework retrieves a recipe for the dish. The recipe is originally in Chinese and Google translated it to English.

tune the probability distribution of ingredients to reduce potential recognition error due to unseen food category.

The remaining sections are organized as follows. Section 3.1 presents the proposed multi-task deep learning models for ingredient recognition, while Section 3.2 introduces the zero-shot recipe retrieval module. Section 3.3 gives detailed introduction on the new food dataset -VireoFood 172. Section 3.4 presents the experimental results for ingredient recognition and zero-shot recipe retrieval. Finaly, Section 3.5 summarizes this chapter.

3.1 Multi-task Deep Learning

The conventional DCNN is an end-to-end system with a picture as the input and the prediction scores of class labels as the output. DCNN models such as AlexNet [35] and VGG [42] are trained under the single-label scenario, specifically, there is

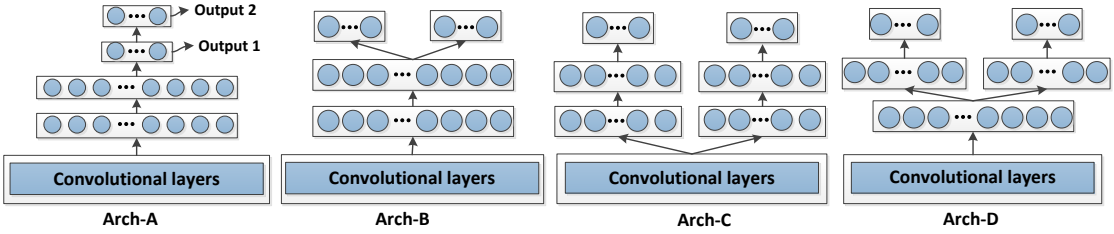


Figure 3.2: Four different deep architectures for multi-task learning of food category and ingredient recognition.

an assumption of exactly one label for each input picture. As ingredient recognition is a multi-label problem, i.e., more than one labels per image, a different loss function needs to be used for training DCNN. On the other hand, directly revising DCNN with an appropriate loss function for ingredient recognition may not yield satisfactory performance, given the varying appearances of an ingredient in different dishes. To this end, we propose to couple food categorization problem, which is a single-label problem, together with ingredient recognition for simultaneous learning.

3.1.1 Architecture Design

We formulate food categorization and ingredient recognition as a multi-task deep learning problem and modify the architecture of DCNN for our purpose. The modification is not straightforward for involvement of two design issues. The first issue is about whether the prediction scores of both tasks should *directly* or *indirectly* influence each other. Direct influence means that the input of one task is connected as the output of another task. Indirect influence decouples the connection such that each task is on a different path of the network. Both tasks influence each other through updating the shared intermediate layers. The second issue is about the degree in which the intermediate layers should be shared. Ideally,

each task should have its own private layer(s) given that the nature of both tasks, single versus multi-labeling, is different. In such a way, the updating of parameters can be done more freely for optimization of individual performance.

Based on the two design issues, we derive four different deep architectures as depicted in Figure 3.2, respectively named as Arch-A to Arch-D. The first design (Arch-A) considers a stacked architecture by placing food categorization on top of ingredient recognition, and vice versa. As the composition of ingredients for different dishes under the same food category can be different, this architecture has the risk that model learning converges slowly as observed in the experiment. The second design (Arch-B) is similar except that indirect influence is adopted and both tasks are at different pathways. Both designs are relatively straightforward to implement by adding additional layers to DCNN. The next two architectures consider the decoupling of some intermediate layers. The third design (Arch-C) allows each task to privately own two intermediate layers on top of the convolutional layers for parameter learning. The last design (Arch-D) is a compromise version between the second and the third architectures, by having one shared and one private layer. Arch-D has the peculiarity that the shared layer can correspond to the high or mid-level features common between two tasks at the early stage of learning, while the private layer preserves the learning of specialized features useful for optimizing the performance of each task.

3.1.2 Implementation

The architectures are modified from the VGG 16-layers network [42]. In terms of design, the major modification is made on the fully connected layers. For the private layers in Arch-D, there are 4,096 neurons for food categorization, and 1024 neurons for ingredient recognition. As food categorization is a single-label

recognition problem, we adopt multinomial logistic loss function L_1 as its loss function; While ingredient recognition is a multi-label recognition problem, cross-entropy is therefore used as the loss function L_2. Denote N as the total number of training images, the overall loss function L is as following:

$$L = -\frac{1}{N}\sum_{n=1}^{N}(L_1 + \lambda L_2) \tag{3.1}$$

where λ is a parameter trading off the loss terms. This loss function is also widely used in other works such as reference [71]. During training, the errors propagated from the two branches are linearly combined and the weights of the first 11 layers shared between two tasks will be updated accordingly. The updating will subsequently affect the last two layers simultaneously, adjust the features separately owned by food and ingredient recognition. Let $\hat{q}_{n,y}$ be the predicted score of image x_n for its ground-truth food label y, L_1 is defined as following:

$$L_1 = \log(\hat{q}_{n,y}) \tag{3.2}$$

where $\hat{q}_{n,y}$ is obtained from softmax activation function. Furthermore, let's denote $p_n \in \{0,1\}^I$, represented as a vector in I dimensions, as the ground-truth ingredients for image x_n. Basically, p_n is a binary vector with entries of value 1 or 0 indicating the presence or absence of an ingredient. The loss function L_2 is defined as

$$L_2 = \sum_{c=1}^{I} p_{n,c}\log(\hat{p}_{n,c}) + (1 - p_{n,c})\log(1 - \hat{p}_{n,c}) \tag{3.3}$$

where $\hat{p}_{n,c}$ denotes the probability of having ingredient category c for x_n, obtained through sigmoid activation function.

3.2 Zero-shot Retrieval

Training a deep network for recognizing all available food categories is not feasible. In addition to the reality that there exist more than tens of thousands of categories, collecting training samples for each of the categories can be a daunting task. Hence, a practical problem is how to leverage the limited knowledge learnt in a network for recognizing dishes of a previously unseen category. As the proposed architectures are capable of predicting ingredients, in principle, the problem can be addressed by retrieving recipes through matching of ingredients. We refer to this problem as zero-shot retrieval, which is to find recipes for test pictures of unseen food categories. Two scenarios are considered here. Suppose each recipe is associated with a picture of the dish. The first scenario is to use the FC7 features, specifically the features extracted from the private layer(s) of Arch-C or Arch-D, to represent images for retrieval. In other words, the search of recipe is equivalent to image retrieval. The second scenario assumes absence of pictures in recipes, and uses the predicted scores of ingredients as the semantic labels for text-based retrieval of recipes. As the approach for the first scenario can be straightforwardly implemented, this section focuses on the presentation of the second scenario. The idea is to incorporate external knowledge to refine the predicted ingredient scores for a more realistic way of zero-shot retrieval.

3.2.1 Ingredient Refinement with CRF

While the composition of ingredients is fuzzy in Chinese food, it is not purely random. Intuitively, certain groups of ingredients co-occur more often (e.g., corn and carrot), while some ingredients are likely exclusive of each other (e.g., fish and beef). Such statistics can be mined from training data and utilized for adjusting the predicting scores of ingredients. Nevertheless, considering the zero-shot problem

and potentially the limited knowledge in deep network, we mine the statistics from a large corpus composed of more than 60,000 Chinese cooking recipes. The major advantage of doing so is to learn a graph modeling ingredient relationships, where their correlations are more generalizable and not restricted by training data, and hence enhance the success rate of zero-shot retrieval.

We extract ingredients from recipes and construct a graph modeling their co-occurrences based on conditional random field (CRF). Let's denote $\mathcal{N} = \{c_1, ..., c_I\}$ as the set of available ingredients and I as its set cardinality. A graph G is composed of the elements of $\mathcal{N}$ as vertices and their pairwise relationships, denoted as $\phi_i(\cdot)$, as edges. Further, let l_i be an indication function that signals the presence or absence of an ingredient c_i. The joint probability of ingredients given the graph is

$$p(l_1, ...l_I) = \frac{1}{Z(\phi)} \exp\Big(\sum_{i,j \in \mathcal{N}} l_i l_j \phi(i,j)\Big) \tag{3.4}$$

where $Z(\cdot)$ is a partitioning function. To learn the graph, we employ Monte Carlo integration to approximate $Z(\cdot)$ and the gradient descent to estimate $\phi(\cdot)$ to optimize the data likelihood [72]. Given a test image, CRF inferences a binary label sequence $\mathbf{y}$ which indicate the occurrence of ingredients based on the graph G. The energy function for inference is composed of unary and pairwise potentials, defined as

$$E(\mathbf{y}) = \sum_{c \in \mathcal{N}} \psi_u(y_c) + \sum_{(c,v) \in \varepsilon} \psi_p(y_c, y_v) \tag{3.5}$$

where ε denotes the set of pairwise cliques. The unary term is set as $\psi_u(y_c) = -\log(x_c)$, where x_c is the predicted score by the deep network for ingredient c. The pairwise potential is defined as

$$\psi(y_u, y_v) = \begin{cases} 0 & \text{if } y_u = y_v \\ \phi(y_u, y_v) & \text{if } y_c \neq y_v \end{cases} \tag{3.6}$$

where the value of $\phi(\cdot)$ is obtained from graph G. Through inferencing, CRF searchs for the optimal label sequence of $\mathbf{y}$ that agrees with the predicted scores and the contextual relationship captured in graph G. We employ an off-the-shelf algorithm, loopy belief propagation [73], for minimizing Eqn-3.5. The output label sequence $\mathbf{y}$ will indicate the presences or absences of ingredients.

3.2.2 Recipe Search

With the output sequence $\mathbf{y}$ by CRF, a query image is represented as a vector $\boldsymbol{Q}^i$. Every element in $\boldsymbol{Q}^i$ corresponds to an ingredient and its value indicates the probability output by CRF. On the other hand, the ingredients extracted from a recipe is represented as a binary vector $\boldsymbol{O}$. The matching score, s_i, between them is defined as

$$s_i = \sum_{c \in \boldsymbol{O} \cap c \in \boldsymbol{Q}^i} x_c \tag{3.7}$$

Note that the score is not normalized in order not to bias recipes with a small number of ingredients. As a result, Eqn-3.7 tends to give a higher score for the recipes with excessive number of ingredients. To prevent such cases, the matching between $\boldsymbol{Q}^i$ and $\boldsymbol{O}$ is performed only for the top-k predicted ingredients with higher probability scores. The value of k is empirically set to 10 as there are few recipes with more than 10 ingredients in our dataset.

3.3 Dataset Collection

We construct a large food dataset specifically for Chinese dishes, namely VIREO Food-172[1], which is made publicly available. Different from other publicly available datasets [10] [11] [12], both food category and ingredient labels are included. In

[1] http://vireo.cs.cityu.edu.hk/VireoFood172/

addition, a large corpus of recipes along with dish pictures is also collected.

3.3.1 VIREO Food-172

The food categories were compiled from "Go Cooking" [2] and "Meishi" [3], which are two websites for popular Chinese dishes. We combine the categories from both websites by removing duplication. All the images in the dataset were crawled from Baidu and Google image search. For each category, the name was issued as keywords in Chinese to search engines. Categories with no more than 100 images returned were removed from the list. For the remaining categories, we manually checked each crawled images up to the depth of 1,300, for excluding images with a resolution lower than 256 $\times$ 256 pixels or suffer from blurring, images with more than one dishes, and false positives. This process ended up with 172 food categories in the dataset.

The 172 categories cover eight major groups of food, as shown in Figure 3.3. The group *meat* contains the most number of categories, with examples including "braised pork" and "sauted shredded pork in sweet bean sauce". On the other hand, there are only eight categories under the group *bean product*, with examples include "Mapo tofu" and "braised tofu". Figure 3.4 shows some examples of food categories in VIREO Food-172.

3.3.2 Ingredient labeling

We compiled a list of more than 300 ingredients based on the recipes of 172 food categories. The ingredients range from popular items such as "shredded pork" and "shredded pepper" to rare items such as "codonopsis pilosula" and "radix

[2]https://www.xiachufang.com/category/

[3]http://www.meishij.net/

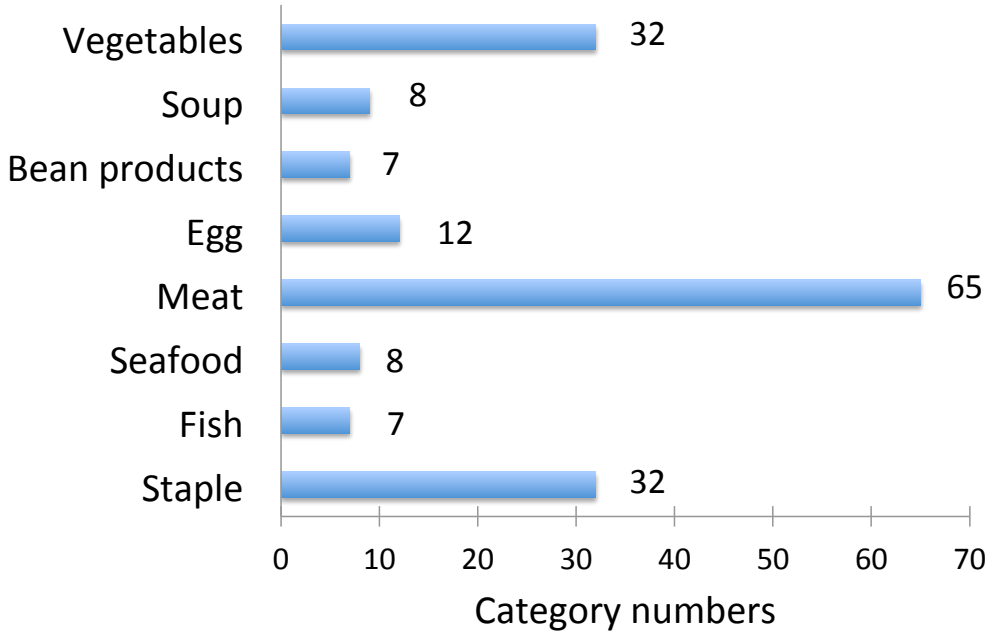


Figure 3.3: The distribution of food categories under eight major food groups in VIREO Food-172.

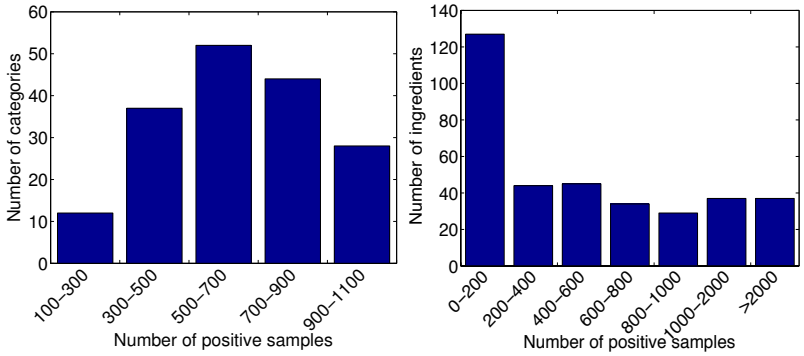
astragali" used for medicinal cooking. Labeling over hundreds of ingredients for over hundred thousands of images could be extremely tedious, not mentioning the challenge of ingredient annotation. Firstly, some ingredients are difficult to be recognized, for example, ingredients under soup or sauce. Secondly, some ingredients are invisible in flour-made food categories such as dumpling and noodle. Thirdly, certain ingredients such as an egg exhibit large visual variations (see Figure 3.5) due to different ways of cutting and cooking. Hence, the labeling considers only the annotation of visible ingredients. In addition, we create additional labels for ingredients with large visual appearance, for example, we have 13 different labels for "egg", such as "preserved egg slices" and "boiled egg".

We recruited 10 homemakers who have cooking experience for ingredient labeling. The homemakers were instructed to label only visible and recognizable ingredients. They were also allowed to annotate new ingredients not in the list,

Figure 3.4: Examples of food categories in VIREO Food-172.

which would be explicitly checked by us. To guarantee the accuracy of labeling, we purposely awarded homemakers with cash bonus as incentives to provide quality annotation, in addition to regular payment. For this purpose, we checked a small subset of labels and provided immediate feedback to homemakers such that they were aware of their performance. The whole labeling process ended in two weeks. By excluding images with no ingredient labels, VIREO Food-172 contains a total of 353 ingredient labels and 110,241 images, with the average of 3 ingredients per image. Figure 3.6 shows the distribution of positive samples in food and ingredient categories. On average, there are 640 positive samples per food category, and 745 per ingredient.

Figure 3.5: The ingredient "egg" shows large difference in visual appearance across different kinds of dishes.



(a) Food category (b) Ingredient

Figure 3.6: The distribution of food categories (a) and ingredients (b).

3.3.3 Recipe Corpus

The corpus was compiled from a popular website "Xinshipu" [4]. The website offers an ontology for 530 key ingredients in Chinese food. Using all of these ingredients as queries, a total of 65,284 Chinese cooking recipes were crawled from this website. Each recipe basically contains four sections, including a brief introduction, ingredient list, cooking procedure, and a picture showing the appearance of the

[4]http://www.xinshipu.com/

dish. The recipes were uploaded by Internet users, and thus there may be multiple recipes sharing the same name but with different ingredient lists. Conversely, there are also few recipes about the same dish but in different names.

3.4 Experiments

We split the experiments into three parts, verifying the performances of multi-task learning (Section 6.1), the impact of CRF (Section 6.2) and the application for zero-shot retrieval (Section 6.3). The first part aims to evaluate different deep architectures for multi-task learning in comparison to single-task DCNN. The last part aims to demonstrate the merit of leveraging ingredient labels for novel recipe retrieval.

3.4.1 Deep Architectures

The experiments are conducted mainly on the VIREO Food-172 dataset. In each food category, 60% of images are randomly picked for training, 10% for validation and the remaining 30% of images for testing. For performance evaluation, the average top-1 and top-5 accuracies are adopted for food categorization, which are standard measures for the single-label task. For ingredient recognition which belongs to multi-label, micro-F1 and macro-F1 that take into account both precision and recall for each ingredient are employed.

The evaluation compares baseline, single and multi-task learnings. The baseline includes SVM classifiers trained using hand-crafted (Gist [74] and color moment [34]) and deep (FC7 of DCNN [35]) features. The single-task learning includes the AlexNet and VGG networks fine-tuned on training and validation sets. Note that for baseline and single-task, different classifiers and networks need to be trained separately for food categorization and ingredient recognition. Specifically, multi-

label SVM (MSVM) is trained for baseline, and cross entropy loss function (Eqn-3.3) is used for single-task DNN. The multi-task learning includes the four deep architectures illustrated in Figure 3.2. Note that we experiment two variants of Arch-A, with the layer of food categorization on top of ingredient recognition (Arch-A1) and vice versa (Arch-A2).

Grid search of parameters is performed to find the best possible model settings for all the compared approaches, based on the training and verification sets. As ingredient recognition involves multiple labels, a threshold is required to gate the selection of labels. The threshold is set to be the value of 0.5 following the standard setting when sigmoid is used as the activation function. For multi-task deep architectures, the learning rate is set to 0.001 and the batch size to 50. The learning rate decays after every 8,000 iterations. Using Arch-D as example, Figure 3.7 shows the impact of the λ parameter in Eqn-3.1. Basically, the Top-1 and Mirco-F1 measures fluctuate within the range of 0.06, when the value of λ varies from 0.1 to 1.0. The best performances attained for food categorization (Top-1) is when $\lambda = 0.1$, and for ingredient recognition (Micro-F1) when $\lambda = 0.3$. To balance the performances, we use F1 of Top-1 and Micro-F1 measures to pick the optimal value, where $\lambda = 0.2$ as shown in Figure 3.7.

Table 3.1 lists the performance of food categorization. The general trend is that deep architectures significantly outperform baselines with either deep or hand-crafted features, while large performance gap is also observed between the results of VGG network and AlexNet. Among the deep architectures for multi-task learning, the designs based on simple modification of DCNN, i.e., Arch-A and Arch-B, show slightly worse performance in Top-1 accuracy compared with single-task VGG. Since the recognition results for both food and ingredients are imperfect, layer stacking as in Arch-A actually could hurt each other's performance. Specifically, the inaccurate prediction in one task will directly affect the other task. On the

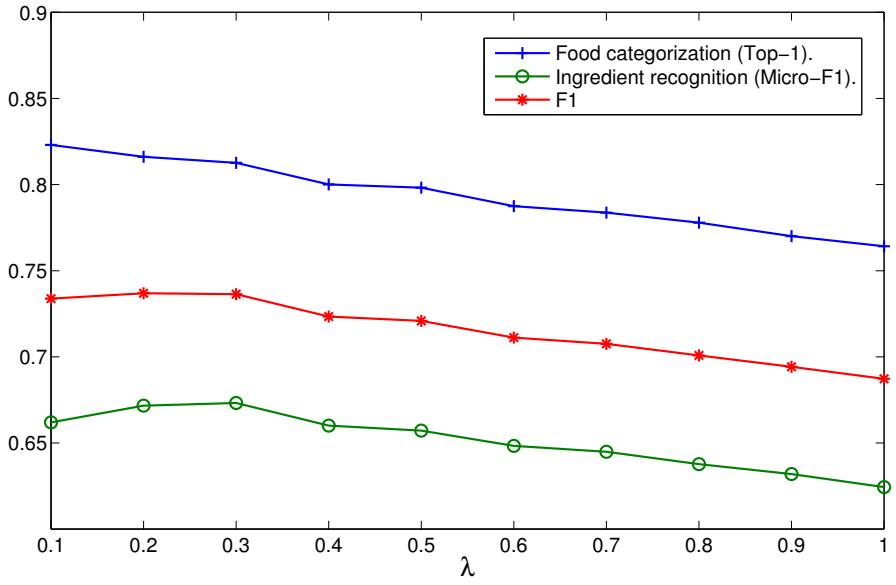


Figure 3.7: Sensitivity of λ parameter in Eqn-3.1 for multi-task deep architecture Arch-D.

other hand, while having separate paths as in Arch-B leads to better performance, the improvement is rather minor by the fact that both tasks share the same lower layers. Basically, the performances of Arch-C and Arch-D show the merit of having separate paths and layers for both tasks. Arch-C, which only shares convolution layers, improves slightly over single-task VGG. We speculate that the design of Arch-C eventually trains two independent layers and hence the advantage over single-task is not obvious. Arch-D, which shares one layer while also learning separate layers tailor-made for different tasks, attains the best performance among all the compared approach for both average Top-1 and Top-5 accuracies.

Table 3.2 shows the performance of ingredient recognition, and similar trends are observed as food categorization. For multi-task learning, all deep architectures except for Arch-A outperform single-task VGG, and with larger performance gaps compared with food categorization. The result basically verifies the merit of joint learning for both tasks. Different from food categorization, sharing layers appears

	Method	Top-1 (%)	Top-5 (%)
Baseline	FC7	48.02	72.01
	Gist	15.39	31.85
	CM	16.54	39.76
Single-task	AlexNet	64.91	85.32
	VGG	80.41	94.59
Multi-task	Arch-A1	78.58	94.24
	Arch-A2	78.63	94.10
	Arch-B	79.05	94.70
	Arch-C	80.66	95.05
	Arch-D	**82.06**	**95.88**

Table 3.1: Average top-1 and top-5 accuracies for single-label food categorization on VIREO Food-172 dataset.

to be a better design choice for ingredient recognition when comparing Arch-B and Arch-C. The best result is attained by Arch-D, which could be viewed as a compromised design between Arch-B and Arch-C. To verify that the improvement is not by chance, we conduct significance test to compare multi-task (Arch-D) and single-task (VGG) using the source code provided by TRECVID[5]. The test is performed by partial randomization with 100,000 numbers of iterations, with the null hypothesis that the improvement is due to chance. At a significance level of 0.05, Arch-D is significantly different from VGG in both food categorization and ingredient recognition by Top-1 accuracy and Macro-F1, respectively. The p-values are close to 0, which rejects the null hypothesis.

To validate the proposed work on other food domains, we also conduct experiments on UEC Food-100 [11] dataset for Japanese dishes. The dataset contains 100 categories of food and totally 14,361 images. Each category has at least 100 positive examples. Nevertheless, ingredient labels are not provided. Similar to VIREO Food-172, we compiled a list of 190 ingredients for Japanese food and conducted

[5]http://www-nlpir.nist.gov/projects/t01v/trecvid.tools/randomization.testing/

Table 3.2: Performance of multi-label ingredient recognition on VIREO Food-172 dataset.

	Method	Micro-F1 (%)	Macro-F1 (%)
Baseline	FC7	42.94	32.22
	Gist	23.01	19.45
	CM	21.08	14.06
Single-task	AlexNet	47.63	34.81
	VGG	60.81	43.73
Multi-task	Arch-A1	55.17	43.75
	Arch-A2	59.69	43.48
	Arch-B	66.32	44.85
	Arch-C	63.44	44.26
	Arch-D	**67.17**	**47.18**

Table 3.3: Performance comparision on UEC Food-100 dataset.

Method	Categorization		Ingredient recognition	
	Top-1 (%)	Top-5 (%)	Micro-F1 (%)	Macro-F1 (%)
FC7	58.03	83.71	52.80	32.51
Gist	30.53	58.80	23.93	11.84
CM	24.11	46.42	16.01	7.830
AlexNet	75.62	92.43	55.62	35.63
VGG	81.31	96.72	57.38	38.62
[38]	78.77	95.15	–	–
Arch-D	**82.12**	**97.29**	**70.72**	**43.94**

manual labeling. A total of 1,317 images are excluded from experiments for no ingredient labels. The experiment is conducted based on 5-fold cross-validation, using the same data split and settings as [38]. In [38], DCNN based on AlexNet is first pre-trained with 2,000 categories in ImageNet, including 1,000 food-related categories. The network is then fine-tuned with training examples in the dataset. Table 3.3 lists the detailed performance. Note that, although not using 1,000 food categories for pre-training, Arch-D still manages to outperform [38] by 3.5% in terms of average top-1 accuracy for food categorization. Overall, similar to the

performance on VIREO Food-172, Arch-D attains the best performances for both tasks.

3.4.2 Effect of CRF

This section verifies the use of CRF in refining the predicted ingredients. All the 65,284 recipes are used for the construction of CRF. A special note is that most recipes do not include the fine-grained description of ingredients. For example, a recipe will simply list "egg" as ingredient, instead of explicitly stating whether the ingredient is either "sliced egg" or "boiled egg". Such information can only be inferred from cooking procedure, such as "leaving the eggs boil for 4 minutes". In this experiment, we do not perform natural language processing to obtain the fine-grained description of ingredients. As a consequence, some labels in VIREO Food-172 are merged and this ends up to 257 ingredient labels for experimentation. For the deep architectures, max pooling is adopted to merge the results of fine-grained ingredients. Specifically, if a network predicts "boiled egg" with the probability of 0.5 and "sliced egg" with 0.1, the probability for "egg" is set to be 0.5 in the CRF.

In addition to assessing the effect of CRF for single and multi-task learnings, we also compare the results against the baseline that directly infers the ingredients from a retrieved recipe. More specifically, given a predicted food category by VGG network, the corresponding recipe is retrieved based on name matching. The predicted labels are then based on the ingredients listed in the recipes. This strategy is often used by some approaches [31] for estimation of nutrition facts. We compare to two baselines, based on the predicted names of food categories or ingredients. Note that a few recipes have the same name despite using different ingredients, and hence multiple recipes could be retrieved. In this case, we only show the result for the recipe which obtains the highest F1 score.

Table 3.4: Ingredient recognition with contextual modeling using CRF.

	Method	*Micro-F1 (%)*	*Macro-F1 (%)*
Baseline (recipe)	Food category	40.75	37.47
	Food ingredient	37.39	33.69
Single-task	Without CRF	63.94	46.81
	With CRF	66.23	48.25
Multi-task	Without CRF	68.84	49.98
	With CRF	**71.25**	**51.18**

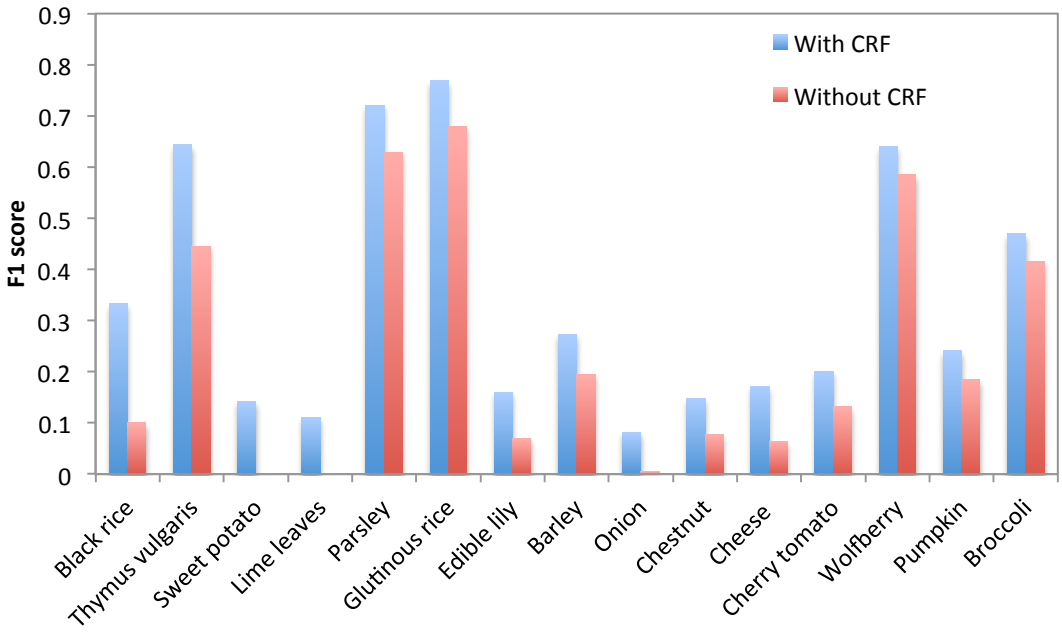


Figure 3.8: The F1 scores of 15 ingredients that achieve large margin of improvement after CRF.

Table 3.4 lists the performance of different approaches. Note that the performance of multi-task is based on Arch-D. Basically, CRF improves the performance of both single and multi-task learnings. All variants of baseline perform poorly in this experiment, far lower than directly using the ingredients predicted by deep architectures. The result is not surprising due to the fact that, for Chinese food, the composition of ingredients for dishes under the same category can vary depending on factors such as geographical regions, weather and culture.

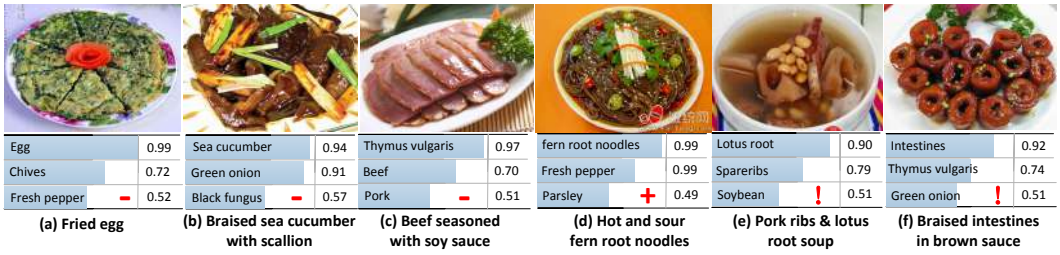


Figure 3.9: Example of test images showing effect of CRF in refining ingredient labels. The "−" sign indicates the false positives that are successfully excluded after CRF, while the "+" sign indicates the false negatives that are recalled by CRF, and the "!" sign indicates true positives that are erroneously removed by CRF.

A few ingredients record large improvement as shown in Figure 3.8. The examples include "black rice" (F1 score = 0.10 to 0.33), "sweet potatoes" (F1 score = 0.00 to 0.14), and "cherry tomato" (F1 score = 0.13 to 0.20). CRF successfully captures the knowledge that "black rice" often co-occurs with "rice" and "soybeans" for food categories involving "cereal porridge". Similarly for the co-occurrence among "cherry tomato", "corn" and "lettuce" for food categories related to "vegetable salad". Figure 3.9 shows a few of success and fail examples refined by CRF. While CRF successfully improves the F1 score and particularly the precision of detection, the recall for few labels also dropped as noticed in Figure 3.9(e) and Figure 3.9(f). Overall, the average precision (micro) is boosted from 0.795 to 0.833, with 193 out of 257 ingredients showing improvement. The average recall (micro) is also boosted from 0.607 to 0.623, with 91 ingredients showing improvement and 88 ingredients dropped.

3.4.3 Zero-shot Recipe Retrieval

This section assesses the use of predicted ingredients for retrieving recipes for food categories unknown in the VIREO Food-172 dataset. We compile a list

of 20 food categories as shown in Table 3.6 for the experiment. Each category is associated with 1 to 20 recipes. For each category, we make sure that at least its key ingredients are known to VIREO Food-172. On average, there are 3 key ingredients per category. Among the 20 categories, 4 out of them include ingredients that are not seen in VIREO Food-172. For each category, a total of 50 images are crawled from Baidu for testing. The experiment is conducted by given a test image, the system searches against 65,284 recipes in the corpus and returns the top-10 recipes. The performance is measured by top-10 hit rate, which counts the percentage of test images where the ground-truth recipes is found in the top-10 rank list.

We compare three major groups of approaches: image retrieval, ingredient matching (Eqn-3.7), and their combination. For image retrieval, only the pictures associated with recipes are involved. We compare the effectiveness of different features for retrieval. For VGG, FC7 feature is extracted from the model trained for ingredient recognition. Similarly for Arch-D, the deep feature is extracted from the private layer specialized for ingredient labels. For ingredient matching, we compare the performances of single (VGG) and multi-task (Arch-D) learnings, where the ingredient prediction scores are both adjusted by CRF. Finally, late fusion is performed for Arch-D and VGG by combining the scores obtained from image retrieval and ingredient matching. Min-max normalization is employed to convert the scores into the range of [0,1]. The fusion is based on joint probability, specifically $1-(1-p_i)(1-p_j)$, where p_i and p_j are scores from different approaches.

Table 3.5 lists the performance of different approaches. For image retrieval, deep features perform significantly better than hand-crafted features. Our proposed model Arch-D outperforms VGG, showing the superiority of multi-task learning not only in recognition but also in feature learning. For text-based ingredient matching, Arch-D also shows better performance than VGG, attributed mainly to the lower recognition error made in ingredient prediction, especially after

Table 3.5: Performance of zero-shot recipe retrieval.

	Method	R@10
Image Retrival	Gist	0.039
	Color moment	0.035
	VGG	0.439
	Arch-D	**0.523**
Ingredient matching	VGG	0.447
	Arch-D (without CRF)	0.462
	Arch-D	**0.554**
Fusion	VGG	0.464
	Arch-D	**0.570**

CRF refinement. Further fusion of both results from Arch-D achieves the overall best performance among all the compared approaches.

Table 3.6 shows the detailed performance of Arch-D on 20 unknown food categories. The performance of image retrieval is influenced by the quality of pictures associated with recipes, particularly for the pictures in low resolution, having different appearances or lighting conditions than the queries. Such examples include "mustard pork noodle" and "tomato & egg noodles". On the other hand, solely matching ingredient lists is limited by the fact that the same set of ingredients can be used for different food categories. One such example is "cucumber & fungus with eggs", where the ingredients are also found in several other food categories, despite different visual appearance due to different ways of cooking and cutting. Image retrieval using the deep features, which are trained to deal with these visual variations, generally shows better performance. Fusion basically compromises both performances and produces the overall best performance. There are four categories where fusion successfully boosts the performances of both approaches. In these cases, image retrieval helps by "disambiguating" the rank lists generated by ingredient matching.

Table 3.6: Recipe retrieval performance on 20 unknown food categories. The number in parentheses indicates the number of recipes for a category. The categories containing unseen ingredients in VIREO Food-172 are indicated by "*".

Category	Image retrival	Ingredient matching	Fusion
Assorted corn (12)	**0.92**	0.84	0.86
Braised noodles with lentil (16)	0.44	0.42	**0.46**
Braised chicken & potato (8)	**0.42**	0.34	0.34
Cucum. & fungus with eggs (2)	**0.68**	0.34	0.56
Carrot & kelp (4)	**0.78**	0.64	0.72
Cabbage & vermicelli (5)	0.30	**0.54**	0.44
Corn, carrot & ribs soup (19)	0.62	0.64	**0.70**
Dried tofu & pepper(8)	0.48	**0.80**	0.68
Griddle cooked chicken*(7)	0.36	0.40	**0.44**
Loofah egg soup (15)	0.76	**0.98**	0.92
Mustard pork noodles (10)	0.34	**0.74**	0.70
Noodles with peas & meat*(4)	**0.30**	0.22	**0.30**
Pepper & bitter gourd (7)	**0.68**	0.60	**0.68**
Ribs claypot (5)	0.16	**0.50**	0.12
Sichuan cold noodles (12)	**0.86**	0.82	0.84
Soybeans & pork leg soup (12)	0.48	0.46	**0.54**
Sausage claypot (19)	0.30	**0.66**	0.60
Spicy crab*(20)	**0.82**	0.38	0.56
Shredded chicken & pea sprouts*(1)	**0.20**	0.08	0.08
Tomato & egg noodles (18)	0.56	**0.94**	0.86

The retrieval performance is also affected by occlusion of ingredients. For example, the "chicken" in "shredded chicken & pea sprouts" is hardly visible under "pea sprouts", which is an ingredient unseen in VIREO Food-172. In this case, ingredient matching performs poorly as seen in Table 3.6. Image retrieval also performs unsatisfactorily due to diverse dish appearances for test images under this category. Another example is "spicy crab", where crab is hidden under other ingredients. Image retrieval, however, performs surprisingly well for this category because of the unique color and texture of the dishes. Finally, there are four

categories that have unseen ingredients. Except for "spicy crab", the performance of these categories is below average, showing the challenges of retrieval for recipes with unknown ingredients.

3.5 Summary

In this chapter, We have presented two main pieces of our work: ingredient recognition and zero-shot recipe retrieval. The former is grounded on a deep architecture (Arch-D) that exploits the joint relationship between food and ingredient labels through multi-task learning. The latter extends the knowledge of Arch-D for the out-of-vocabulary scenario, by learning contextual relationships of ingredients from a large textual corpus of recipes. Experimental results on a challenging Chinese food dataset (VIREO Food-172) show that, while the performance of food categorization is enhanced slightly, the improvement in ingredient recognition is statistically significant compared to the best single-task VGG model. The superiority in performance is not only noticed in VIREO Food-172 but also in UEC Food-100, a large-scale Japanese food dataset. More importantly, when extracting the deep features (FC7) from the specialized or private layer learnt for ingredient recognition, the features show highly favorable performance for zero-shot recipe retrieval, in comparison to hand-crafted features and single-task model. The performance of ingredient recognition is also successfully enhanced with the contextual relationship modeling of ingredients and CRF. The experiment also indicates that using our proposed architecture and CRF for ingredient prediction can produce better performance than directly inferring ingredients from recipes searched by VGG. When further using the predicted ingredients for matching recipes of unknown food categories, our model also demonstrates impressive performance, including when fusing with the deep features.

While encouraging, the current work is worth further investigation in two directions. First, cooking method (e.g., frying, steaming, grilling) is not explicitly considered in the developed deep architecture. In the experiment, we notice that some dishes have the same ingredients but appear visually different mainly due to different cooking methods. Our current approach basically cannot distinguish recipes for this kind of dishes. Similarly for ways of cutting ingredients (e.g., chop, slice, mince) which may demand a hierarchical way of ingredient recognition in deep network. In addition, our multi-task model could not deal with ingredients (e.g., honey, soybean oil) that are not observable or visible from dishes. Secondly, while this chapter considers the zero-shot problem of unknown food categories, how to couple this problem together with unseen ingredients remains unclear. Future work may include learning of embedded space that can capture the inherent "translation" between dish pictures and textual recipes, for dealing with the problem of unknown food and ingredient labels.

CHAPTER 4

RICH ATTRIBUTE LEARNING FOR CROSS MODAL RECIPE RETRIEVAL

Similar in spirit as the previous chapter, this chapter performs fine-grained ingredient recognition for searching of cooking recipes. Particularly, we address the problem in real-world that there are many different varieties of dishes cooked with the same ingredients. Hence, recognition using ingredients alone is inherently insufficient to retrieve recipes. Figure 4.1 shows three different examples of dishes that use the same ingredients. Basically, different cutting methods result in different shape appearances, for example in Figure 4.1(a), the shredding or slicing of green pepper and potato alters the outlook of dishes. Similarly, different cooking methods, such as "pan-fry" and "stir-fry" shown in Figure 4.1(b), can change the colour and texture appearance of the dishes. When the methods for both cutting and cooking are different, the appearance can be wildly diverse as the fish dishes shown in Figure 4.1(c). Hence, we argue that effective food recognition generally requires knowledge of cutting and cooking attributes beyond ingredients. From the viewpoint of "healthy eating", knowing cooking attributes also provide helpful clues for nutrition analysis. For example, "boiling" can wash away water-soluble vitamins, and people with diabetes should limit the intake of "deep-fried" food.

Technically, we can tackle the aforementioned problem by directly embedding the cutting and cooking methods into ingredient labels. For example in the previous chapter, there were 13 labels used for the "egg" ingredient to characterize different ways of cutting and cooking an egg. Such labeling strategy, although

(a) cutting (b) cooking (c) cutting + cooking

Figure 4.1: Examples of dishes with the same ingredients but different cutting and cooking methods.

practically useful for ingredient recognition, is difficult to scale up due to the exponential number of attribute combinations for the ingredient, cutting and cooking methods. In this chapter, we consider as many as 1,276 ingredients, 8 cutting and 36 cooking methods. By brute force combination of these attributes, there could be close to 0.4 million labels; this is beyond the capacity that a deep neural network can be trained, unless with sufficient training samples and machines.

This chapter proposes the retrieval of recipes by rich food attributes, i.e., ingredients, cutting and cooking methods. The three attributes are recognized by a three-way deep architecture, which are learnt in a multi-task manner. Specifically, each task aims to predict a particular type of attribute, while shared visual features are learnt to minimize the overall prediction error of the three attributes. A peculiar challenge is that the prediction of cutting and cooking attributes requires the knowledge of ingredient locations. In other words, ingredient regions have to

be localized as the recognition of attributes should happen at the image region level. This basically makes the design of deep neural architecture fundamentally different from [21]. In our proposal, the localization of ingredients is learnt in an unsupervised manner without the requirement of region-level training information. In addition, to keep track of the localized correspondence among the three types of attributes, a new pooling technique is tailored to combine the results of prediction from different tasks.

The remaining sections are organized as follows. Section 4.1 presents multi-task deep architecture for rich attribute learning. Section 4.2 describes the retrieval of recipes based on the prediction of multi-task model. Finally, Section 4.3 presents experimental results, and Section 4.4 summarizes this chapter.

4.1 Rich Attribute Learning

Framework. Figure 4.2 presents an overview of the proposed framework. Given a food picture I, a pyramid of multi-resolution images is generated and input to a deep convolutional network (DCNN). The corresponding feature maps are extracted from DCNN and subsequently transformed into embedded features for the prediction of ingredients, cutting and cooking methods. The attributes are pooled across image regions and scales before being matched against text recipes for retrieval.

Feature embedding layer. Our DCNN architecture uses the VGG network [42]. The $Pool5$ feature maps, which correspond to the last convolution layer of DCNN and retain the spatial information of the original image, are extracted from VGG for feature embedding. The $Pool5$ feature is divided into $m \times m$ grids, where each grid is represented by a vector of 512 dimensions. The value of m varies depending on the image size. For an image of size 448×448, $m = 14$ and

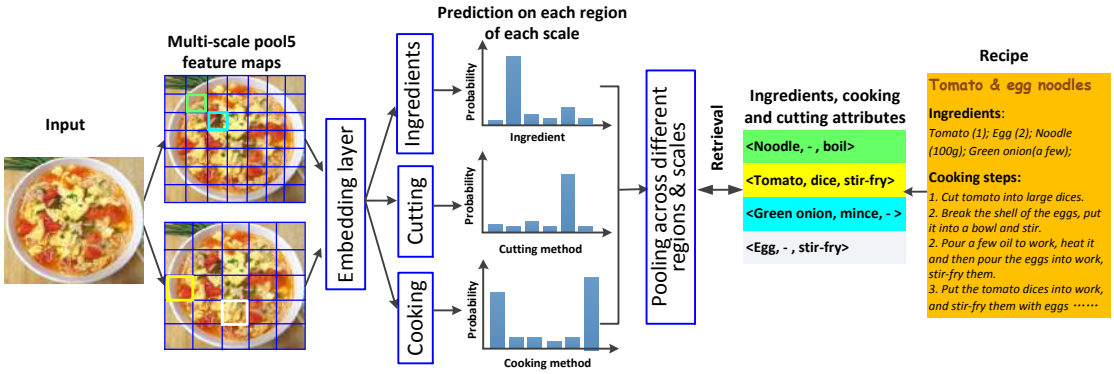


Figure 4.2: Cross-modal recipe retrieval: The ingredients, cutting and cooking methods extracted from an image query are matched against the information derived from the textual recipe.

each grid corresponds to a receptive field of 32×32 resolution. We denote $\boldsymbol{F}_I$ as the $Pool5$ feature and its grids or regions as $\boldsymbol{f}_i$, where $i \in [0, m \times m]$. Each region $\boldsymbol{f}_i$ is transformed to an embedding feature as follows:

$$\boldsymbol{v}_i = \tanh(\boldsymbol{W}_I \boldsymbol{f}_i + \boldsymbol{b}_I) \tag{4.1}$$

where $\boldsymbol{v}_i \in \mathbb{R}^d$ is the transformed vector in a d-dimensional space, $\boldsymbol{W}_I \in \mathbb{R}^{d \times 512}$ is the learnt transformation matrix and $\boldsymbol{b}_I \in \mathbb{R}^d$ is the bias term.

Region-wise multi-task classification. The predictions of attributes are learnt simultaneously in a multi-task manner, taking advantage of their joint relationships in modeling the diverse dish appearances. Specifically, while each prediction is regarded as a separate task, they share the same feature embedding layer which is learnt to optimize the predictions of three different attribute types. Notice that the learning is conducted region-wise and the prediction is made directly on each grid of an image.

As each grid depicts a small region of the original image, a reasonable assumption being adopted is that there is one dominant ingredient per region. Furthermore, an ingredient is assumed to undergo at most one cutting procedure and one

dominant cooking procedure. There are a few cases where this assumption is violated, such as in the example, shown in Figure 4.2, where the egg and tomato are stir-fried after being boiled. In this case, we consider "stir-fry" as the dominant cooking method for altering the appearance of egg and tomato. Therefore, the predictions of ingredient, cutting and cooking labels for each region are regarded as single-label classification.

The activation function being applied is softmax for getting the probability distributions of ingredient, cutting and cooking labels, denoted as $\hat{\boldsymbol{p}}_{\text{ingre},i} \in \mathbb{R}^t$, $\hat{\boldsymbol{p}}_{\text{cut},i} \in \mathbb{R}^c$ and $\hat{\boldsymbol{p}}_{\text{cook},i} \in \mathbb{R}^k$ respectively, for the i-th region as follows:

$$\hat{\boldsymbol{p}}_{\text{ingre},i} = softmax(\boldsymbol{W}_{\text{ingre}}\boldsymbol{v}_i + \boldsymbol{b}_{\text{ingre}}), \tag{4.2}$$

$$\hat{\boldsymbol{p}}_{\text{cut},i} = softmax(\boldsymbol{W}_{\text{cut}}\boldsymbol{v}_i + \boldsymbol{b}_{\text{cut}}), \tag{4.3}$$

$$\hat{\boldsymbol{p}}_{\text{cook},i} = softmax(\boldsymbol{W}_{\text{cook}}\boldsymbol{v}_i + \boldsymbol{b}_{\text{cook}}), \tag{4.4}$$

where t, c and k denote the number of ingredients, cutting and cooking labels respectively. The learnt transformation matrices are $\boldsymbol{W}_{\text{ingre}} \in \mathbb{R}^{t\times d}$, $\boldsymbol{W}_{\text{cut}} \in \mathbb{R}^{c\times d}$ and $\boldsymbol{W}_{\text{cook}} \in \mathbb{R}^{k\times d}$, and similarly for the bias terms $\boldsymbol{b}_{\text{ingre}} \in \mathbb{R}^t$, $\boldsymbol{b}_{\text{cut}} \in \mathbb{R}^c$ and $\boldsymbol{b}_{\text{cook}} \in \mathbb{R}^k$.

Region-level dependency pooling. Since each region is associated with the probability distributions of three different attributes, a straightforward way to obtain the image-level labels is by an independent max-pooling over the three distributions across regions. Nevertheless, such a simple scheme overlooks the joint relationship among the three attributes. For example, a region that contributes to the response of an ingredient does not guarantee that its cooking and cutting attributes will be counted during max pooling. As a result of independent pooling, the three image-level attributes could be inconsistent which could confuse and complicate the learning of network parameters.

We propose dependency pooling by first performing max pooling of ingredient labels across regions, followed by pooling of cutting and cooking attributes only from the regions that contribute most to the image-level ingredient labels. Let $\hat{\boldsymbol{p}}_{\text{ingre},I}$ be the probability distribution of ingredients for image I. The response of an ingredient indexed by the j-th element is obtained as follows:

$$\hat{\boldsymbol{p}}_{\text{ingre},I}(j) = \max\{\hat{\boldsymbol{p}}_{\text{ingre},i}(j)|_{i=1}^{m^2}\} \tag{4.5}$$

where m^2 is the total number of image grids. For each ingredient, the i-th region which contributes most to the response will be tracked. Subsequently, the $\hat{\boldsymbol{p}}_{\text{cut},i}$ and $\hat{\boldsymbol{p}}_{\text{cook},i}$ of every region will be pooled or aggregated to form two matrices, denoted as $\hat{\boldsymbol{P}}_{\text{cut},I} \in \mathbb{R}^{c\times t}$ and $\hat{\boldsymbol{P}}_{\text{cook},I} \in \mathbb{R}^{k\times t}$ respectively. In other words, each $\hat{\boldsymbol{P}}_{\text{ingre},I}(j)$ is indexed to vectors $\hat{\boldsymbol{P}}_{\text{cut},I}(j)$ and $\hat{\boldsymbol{P}}_{\text{cook},I}(j)$, corresponding to the prediction of cutting and cooking attributes for ingredient j.

Loss function. The loss function is defined for each attribute type and then linearly averaged as follows:

$$L = \frac{1}{N}\sum_{n=1}^{N}(L_1 + L_2 + L_3) \tag{4.6}$$

where L_i is a loss function referring to the prediction of ingredient (L_1), cutting (L_2) or cooking (L_3), and N is the total number of training images. Note that, as L_1 is calculated at the image-level, ingredient recognition is a multi-label classification problem. The loss functions L_2 and L_3, on the other hand, are calculated on the basis of every ingredient. As each ingredient is associated to one cooking and one cutting method, L_2 and L_3 are all single-label classification problems. The loss function used for L_1, L_2, and L_3 is cross-entropy. We denote $\boldsymbol{p}_{\text{ingre},I_n} \in \{0,1\}^t$ as the ground-truth ingredients for a food picture I_n, represented by a binary vector whose elements are either 1 or 0 indicating the presence or absence of a particular

ingredient. The loss function L_1 is defined as

$$L_1 = \sum_{\jmath=1}^{t} (\boldsymbol{p}_{\text{ingre},I_n}(\jmath) \log(\hat{\boldsymbol{p}}_{\text{ingre},I_n}(\jmath)) + (1 - \boldsymbol{p}_{\text{ingre},I_n}(\jmath)) \log(1 - \hat{\boldsymbol{p}}_{\text{ingre},I_n}(\jmath))) \tag{4.7}$$

Furthermore, we denote $\boldsymbol{a} = \{a, \boldsymbol{p}_{\text{ingre},I_n}(a)|_{a=1}^{t} = 1\}$ be the ingredients visible in I_n. For each ingredient $a \in \boldsymbol{a}$, let $\boldsymbol{P}_{\text{cut},I_n}(a) \in \{0,1\}^c$ as its ground-truth cutting label and $\boldsymbol{P}_{\text{cook},I_n}(a) \in \{0,1\}^k$ as its ground-truth cooking label. The loss functions L_2 and L_3 are defined as follows:

$$L_2 = \sum_{a \in \boldsymbol{a}} \sum_{\nu=1}^{c} (\boldsymbol{P}_{\text{cut},I_n}(a,\nu) \log(\hat{\boldsymbol{P}}_{\text{cut},I_n}(a,\nu)) + (1 - \boldsymbol{P}_{\text{cut},I_n}(a,\nu)) \log(1 - \hat{\boldsymbol{P}}_{\text{cut},I_n}(a,\nu))) \tag{4.8}$$

$$L_3 = \sum_{a \in \boldsymbol{a}} \sum_{\mu=1}^{k} (\boldsymbol{P}_{\text{cook},I_n}(a,\mu) \log(\hat{\boldsymbol{P}}_{\text{cook},I_n}(a,\mu)) + (1 - \boldsymbol{P}_{\text{cook},I_n}(a,\mu)) \log(1 - \hat{\boldsymbol{P}}_{\text{cook},I_n}(a,\mu))) \tag{4.9}$$

During training, the errors accumulated from three tasks are back-propagated through the network till the embedding layer. The involved parameters, including the shared (e.g., $\boldsymbol{W}_I$) and dedicated (e.g., $\boldsymbol{W}_{\text{ingre}}$, $\boldsymbol{W}_{\text{cut}}$, $\boldsymbol{W}_{\text{cook}}$) parameters, will be updated correspondingly to simultaneously optimize the recognition performances of the three tasks.

Multi-scale recognition. The size of an ingredient varies depending on factors such as the cutting methods and camera-to-dish distance. Using a fixed resolution of grids cannot handle the change in scale. The problem is tackled by the generation of pyramid images in multiple resolutions. In this way, the receptive field of a grid can spatially extend to a larger scope depending on the resolution of the input image. For example, for an image of size 448×448 pixels, each grid in the

*pool*5 feature map corresponds to a receptive field of a 32×32 pixels image region. By reducing the size of image to a resolution of 224×224 pixels, the receptive field extends to the spatial size of 64×64 pixels in the original image before resizing.

The consideration of multi-scale recognition will only introduce a minor change to the network architecture. Except for region-level pooling that involves consolidation of attribute predictions from multiple scales, the updating of parameters remains the same throughout the learning procedure. By denoting $\boldsymbol{p}^l_{\text{Ingre},I}$ as the probability distribution of ingredients at scale l, max pooling is conducted across different regions and scales as follows:

$$\hat{\boldsymbol{p}}_{\text{ingre},I}(j) = \max\{\max\{\hat{\boldsymbol{p}}^l_{\text{ingre},i}(j)|_{i=1}^{m^2}\}|_{l=1}^{L}\}. \tag{4.10}$$

During pooling, the regions which contribute most to the response of a particular ingredient are tracked for aggregation of $\hat{\boldsymbol{P}}_{\text{cut},I}$ and $\hat{\boldsymbol{P}}_{\text{cook},I}$. Basically, the multi-scale design ensures that an ingredient and its cooking and cutting attributes can be adaptively pooled from a region in a particular scale that exhibits the highest possible prediction confidence.

4.2 Cross-modal Recipe Retrival

Using the proposed deep architecture, a query picture Q is represented by $\hat{\boldsymbol{p}}_{\text{ingre},Q} \in \mathbb{R}^t$, $\hat{\boldsymbol{P}}_{\text{cut},Q} \in \mathbb{R}^{c \times t}$ and $\hat{\boldsymbol{P}}_{\text{cook},Q} \in \mathbb{R}^{k \times t}$, corresponding to the probability distributions of ingredients, cooking and cutting methods respectively. On the other hand, a text-based recipe R is represented by a set of attribute triplets, $\{< x, \text{cut}_x, \text{cook}_x > | x \in O\}$, where O is the set of ingredients extracted from R, cut_x denotes the cutting attribute for ingredient x and similarly for cook_x. The

similarity between Q and R is defined as

$$\mathrm{Sim}(Q,R) = \frac{\sum_{x\in O}(\hat{\boldsymbol{p}}_{\mathrm{ingre},Q}(x) + \lambda(\hat{\boldsymbol{P}}_{\mathrm{cut},Q}(x,\mathrm{cut}_x) + \hat{\boldsymbol{P}}_{\mathrm{cook},Q}(x,\mathrm{cook}_x)))}{|O|}, \tag{4.11}$$

The parameter λ denotes the importance of cutting and cooking attributes, where its value is learnt to be 0.2 on the validation set. The similarity score is normalized in order not to erroneously favouring recipes that contain an excessive number of ingredients.

4.3 Experiments

4.3.1 Dataset

There are a number of public food datasets, such as UEC Food-100 [11], Food-101 [10], VIREO Food-172 [21], but none of them includes the cooking and cutting attributes of ingredients. We therefore constructed a new food dataset, by crawling 47,882 recipes along with images associated with these recipes from the "Go Cooking" [1] website. The dataset is composed of mostly Chinese food, ranging from regular dishes, snacks and desserts to Chinese-style western food. We recruited 10 homemakers who have cooking experience for attribute labeling. A total of 1,276 ingredients compiled from the recipes are labeled. The lists of cutting and cooking attributes are shown in Table 4.1, which are respectively compiled from "Go cooking" and "meishijie" [2] websites.

During manual annotation, a homemaker was provided with recipes along with food images. The homemaker was first instructed to read a recipe so as to understand the cooking procedure. The list of ingredients extracted from the recipe

[1] http://www.xiachufang.com/

[2] http://so.meishi.cc/index.php?q=cooking/

was then prompted to homemaker for selection of visible ingredients in a food picture. Ingredients missing from the recipe can also be manually input. As some ingredients are named together with cooking and cutting methods in the recipe, we manually reverted these ingredients to their original names before prompting to homemakers for selection. For example, the ingredient "seasoned beef slice" was reverted to "beef". The homemakers were also requested to also label the cutting and cooking methods of each ingredient from the provided lists as shown in Table 4.1. Input of "null" label for cutting and cooking method was allowed when the ingredient was not subject to any cutting or cooking method. To guarantee the quality of annotation, we sampled a subset of labels for manual checking and provided verbal feedbacks to homemakers whenever labeling was inaccurate or imprecise. The whole annotation procedure ended in about two weeks. On average, each food picture contained three visible ingredients. Furthermore, each ingredient had on average 121 positive training samples.

Table 4.1: List of cutting and cooking methods

Cut	Batonnet	Brunoise dice	Large Dice
	Mince	Roll cut	Shreds
	Slice	Smash	
Cook	Baking	Bake in pan	Braising
	Brewing	Bake stewing	Boiling
	Braising with starch	Clay pot cooking	Casserole
	Cover and simmer	Deep frying	Dressing
	Extreme-heat stir-fry	Food drying	Flash-frying
	Gratinated	Griddle cooking	Grilling
	Gradual simmering	Hot candied	Jellying
	Juicing	Moist stir-fry	Microwaving
	Pan-frying	Pickling	Quick-frying
	Quick boiling	Roasting	Scalding
	Seasoned with soy sauce	Steaming	Sugar dipped
	Slow red cooking	Stir-frying	Smoking

4.3.2 Experimental setting

Instead of learning a deep model from scratch, we extend the VGG architecture trained on VIREO Food-172 [21]. The model was reported to exhibit fairly good recognition performance on 353 ingredients. We extend the model to 1,276 ingredients and plug in two additional tasks for the recognition of cooking and cutting labels. The dataset is split into three sets: 80% for training, 10% for validation, and 10% for testing.

During training, the dimension of the embedding feature (Equation-1) is set to $d = 300$ and validated to be effective on the validation set. Two-level of pyramid images, respectively at resolutions of 448×448 pixels and 224×224 pixels, are used for multi-scale recognition. The model is trained using stochastic gradient descent with the momentum set as 0.9. The initial learning rate is set to be 0.1 and the batch size is 50. The learning rate decays after every 3,000 iterations. Finally, as each food picture only has a small number of ingredients out of the available 1,276 ingredients, the ground-truth vector P_{ingre,I_n} is very sparse. As a result, negative sampling is adopted by randomly selecting 10% of negative samples for training.

4.3.3 Recognition performance

As ingredients involve multiple labels, a threshold is required to gate the selection of labels. The threshold is set to 0.5, following the common practice for deep

Table 4.2: Food attribute prediction at different scales

	Ingredient			Cutting	Cooking
Scale	Recall	Precision	F1	Accuracy	Accuracy
Single (448×448)	0.369	0.233	0.286	0.601	0.442
Single (224×224)	0.260	**0.252**	0.256	0.580	0.427
Multi ($448 \times 448 + 224 \times 224$)	**0.391**	0.240	**0.297**	**0.623**	**0.461**

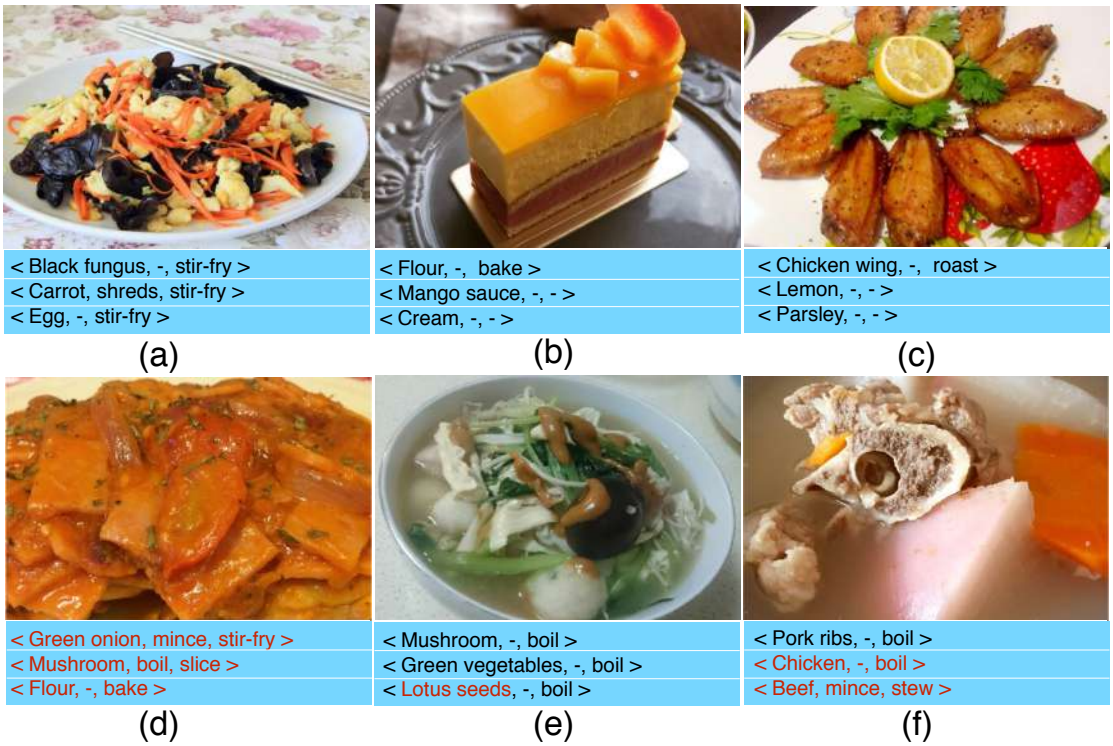


Figure 4.3: Examples of attribute prediction. False positives are marked in red. The sign "-" indicates no cutting or cooking method is applied.

learning based multi-label recognition. In other words, only labels whose prediction scores surpass the threshold are considered as being recognized. Table 4.2 shows the performance of our proposed multi-task model. As recognition of cutting and cooking methods belong to single-label classification, only top-1 accuracy is shown. As ingredients are cut into small pieces in most of the dishes, the use of higher resolution (448×448) images achieves better recall in ingredient recognition and accuracy in prediction of cooking and cutting methods. On the other hand, lower resolution (224×224), which has larger receptive field and hence can consider more surrounding context of ingredients, obtains higher precision. By combining both scales, the best performances are attained for all the three attributes. Figure 4.3 shows some examples of attribute predictions. In general, the results are satisfactory except in several cases which are summarized as follows.

Table 4.3: Ingredient recognition: multi versus single task learning

Task	Ingredient recognition		
	Recall	Precision	F1
Ingredient (FC) [21]	0.121	0.110	0.115
Ingredient (Region)	0.358	0.216	0.269
Ingredient + cutting	0.381	0.236	0.291
Ingredient + cooking	0.373	0.234	0.288
Ingredient + cutting + cooking	**0.391**	**0.240**	**0.297**

First, the model is limited in recognizing ingredients with similar shape and color. For example, in Figure 4.3(e), "fish ball" is incorrectly predicted as "lotus seed". Both ingredients appear somewhat similar when cooked in soup. More typical examples are indeed different kinds of meat (e.g., "duck" and "chicken") which could have similar texture pattern when being cooked by certain methods like "stewing" or "braising". In Figure 4.3(f) for example, "pork rib" is wrongly recognized as "beef" and "chicken". Despite these failure examples, our model can always correctly recognize the ingredients with unique shape, such as the "chicken wing" in Figure 4.3(c). Second, our model always fails to predict ingredients covered under sauces such as that shown in Figure 4.3(d). Such examples are not easy to be recognized even by the human. Third, as we consider only two levels of pyramid images, our model cannot deal with pictures with close-up view of dishes, such as that in Figure 4.3(f). In this example, both "carrot" and "white radish" are identified incorrectly due to the limited scope of receptive field. Fourth, certain cutting attributes are relatively hard to be distinguished, for example "roll cut" and "large dice", where the former (latter) cuts ingredients into pieces with two angled sides (blocks). The cutting effect is not easily observed especially when ingredients are mixed or occluded with each other. Comparatively, cutting methods such as "shred", "mince" and "slice" which result indistinguishable shapes of ingredients can always be predicted correctly. Similarly for cooking attributes,

Figure 4.4: The ingredient "flour" appears wildly diverse under different cooking methods but still can be recognized by our model.

where the effect of "gradual simmering", "cover and simmer" and "bake stewing" are not visually distinguishable. Finally, the correlation between the ingredient and cooking attributes also affects the prediction accuracy. For example, the outlooks of certain ingredients such as "carrot", "black fungus" and "peas" change a little despite undergoing different cooking methods. As a result, while these ingredients are relatively easy to recognize, their cooking attributes are often predicted wrongly. On the other hand, there are ingredients which associate only with few cooking methods, for example "flour" and "baking". The multi-task model always yields high prediction accuracy for these attributes.

We also compare our model with the deep architecture in [21], which reported state-of-the-arts results for ingredient recognition on UEC Food-100 and VIREO Food-172 datasets. The architecture has two pathways, one for food categorization and the other for ingredient recognition. We take away the pathway for food categorization and fine-tune the network with 1,276 ingredients in our dataset. Additionally, as [21] learns embedding features from FC (fully-connected) layer of DCNN, we also compare to a variant of the model which learns features from $Pool5$ convolutional layer and performs region-level prediction and pooling for ingredient recognition.

Table 4.3 lists the performance of three different approaches. Basically, our

proposed model, which is composed of three pathways, outperforms the other two single task models. Besides, learning features from convolution layer (region) performs significantly better than FC layer. As ingredient is generally small in size, region-level recognition is superior to image-level. We attribute the success of our model to the fact that, by having cutting and cooking information, our model has a better capability in dealing with diverse appearances of the ingredient. Cutting attribute contributes to larger improvement than cooking for enjoying higher prediction accuracy as shown in Table 4.3. Figure 4.4 shows examples where the ingredient "flour" appears to be varied in different dishes but can still be correctly recognized by our model and not by [21], either with FC or region. To verify that the improvement is not by chance, we conduct significance test to compare our architecture with both single task models. Using the source code provided by TRECVID[3], the test is performed by partial randomization with 100,000 number of iterations for F1 measure. At the significance level of 0.05, our architecture is significantly better than other models with p-values close to 0, which rejects the null hypothesis that the improvement is by chance.

4.3.4 Recipe retrival

We compile a list of 1,000 images from the test set as queries for cross-modal recipe retrieval. Each query has only one ground-truth recipe. These queries are searched against a dataset composed of 4,985 recipes. Among them, 1,278 of recipes share exactly the same set of ingredients with at least another one recipe, despite belonging to different dishes. We purposely select the queries such that there are 412 of them whose ground-truth recipes are a subset of 1,278 recipes. Furthermore, in order to verify the advantage of cutting and cooking attributes, only images whose F1>0.3 in ingredient recognition for all the three approaches

[3]http://www-nlpir.nist.gov/projects/t01v/trecvid.tools/randomization.testing

Table 4.4: Contribution of different attributes to recipe retrieval. The best performance is highlighted in bold font.

Method	MRR	R@1	R@5	R@10	R@20
Ingredient	0.128	0.169	0.284	0.436	0.585
Ingredient + cut	0.141	0.185	0.320	0.483	0.704
Ingredient + cook	0.133	0.171	0.286	0.473	0.698
Ingredient + cut + cook	**0.153**	**0.204**	**0.338**	**0.518**	**0.742**

shown in Table 4.3 are selected as queries. The following metrics are employed for performance evaluation.

- Mean reciprocal rank (MRR): MRR measures the reciprocal of rank position where the ground truth recipe is returned, averaged over all the queries. This measure assesses the ability of the system to return the correct recipe at the top of the ranking. The value of MRR is within the range of [0, 1]. A higher score indicates a better performance.

- Recall at Top-K (R@K): R@K computes the fraction of times that a correct recipe is found within the top-K retrieved candidates. R@K provides an intuitive sense of how quickly the best recipe can be located by investigating a subset of the retrieved items. As MRR, a higher score also indicates a better performance.

Table 4.4 shows the incremental improvement in retrieval when cutting and cooking methods are incorporated. Cutting attributes basically introduce a higher degree of improvement than cooking methods across all the measures. This is mainly because the prediction of cutting attributes is more accurate. The best result is attained when all the three attributes are jointly considered. We conduct significance test to verify the improvement is not by chance. Using partial randomization, the test suggests that there is a significant difference between using

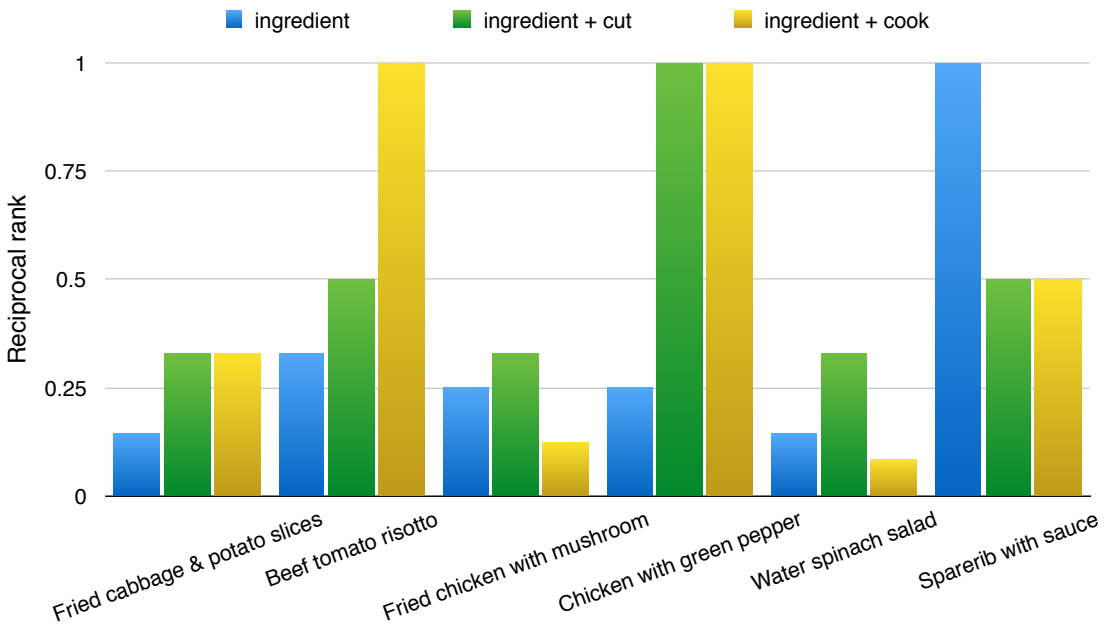


Figure 4.5: The impacts of different attributes on 6 query examples.

all three attributes and using ingredient attribute only at the level of 0.05.

Figure 4.5 compares the reciprocal rank scores of using different attributes for recipe retrieval on 6 query examples. Different attributes basically have different impacts on different dishes. For example, cooking attribute improves the retrieval performance on the query "Beef tomato risotto" but decreases the performance on "water spinach salad". Similarity, cutting attribute boosts the retrieval performance of "Fried cabbage & potato slices" but decreases the performance of "Sparerib with sauce". Basically, when the dish contains popular ingredients, like potatoes, tomatoes that can be cut/cook in different ways for preparing different dishes, cooking/cutting attributes will be extremely useful for boosting retrieval performance as long as those attributes can be correctly identified.

Figure 4.6 shows three examples of recipe retrieval. In Figure 4.6(a), the two recipes ranked at the top contain the same set of ingredients. By correctly recognizing the cooking methods of ingredients, our model successfully ranks the

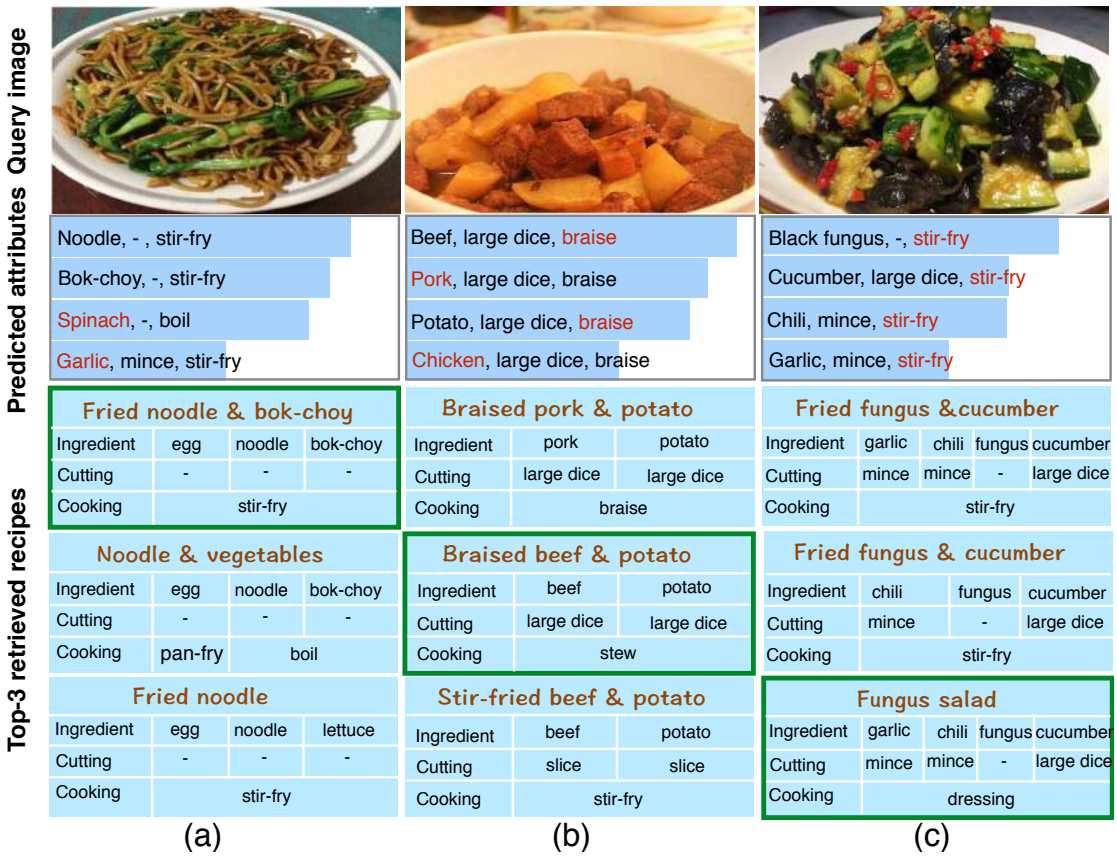


Figure 4.6: Examples of top-3 retrieved recipes (ranked from top to bottom). Ground-truth recipe is marked in green.

ground-truth recipe at the top. Figure 4.6(b) shows an example where one of the key ingredients ("pork") and its cooking attribute are predicted wrongly. However, by correctly recognizing cutting attributes of key ingredients, our model is still able to rank the ground-truth recipe higher than the recipe with the same ingredients but different cutting and cooking methods. Figure 4.6(c) shows an example where our model cannot distinguish recipes with similar ingredients where cooking attributes are predicted wrongly.

Next, we compare our architecture with three other approaches. The first is a multi-task model in [21], where we take the pathway for ingredient recognition and fine tune with 1,274 ingredient labels in our dataset. We term the method

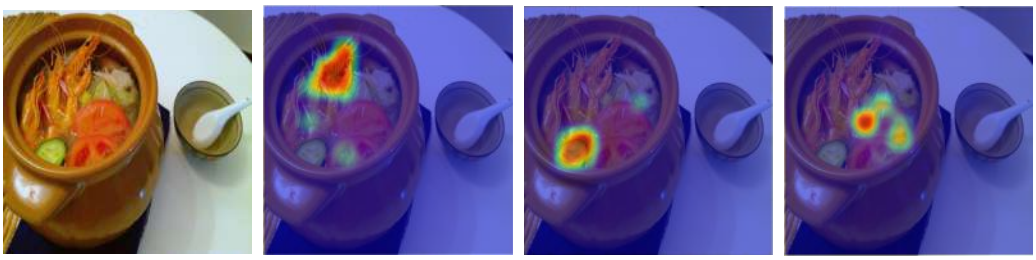
as "single-task". As [21] embeds cooking and cutting attributes directly into the ingredient labels, the second approach takes the same strategy. We combine the three different attributes by brute-force, resulting in 20,736 labels with training examples. By further removing labels with less than 10 examples, we only manage to retain 1,345 labels, which are used to fine-tune the ingredient pathway in [21]. We term the second method as "single-task (BF)". In the experiment, we implement two versions of these approaches, by extracting features from fully-connected (FC) and convolutional (region) layers of DCNN. The third approach is based on the attention model recently proposed in [23], which learns a joint feature representation between visual and text by stacked attention model [75]. We train the model using the same training and validation sets as our proposed architecture. Table 4.5 lists the result of comparison. As expected, the brute-force combination of different attributes leads to better performance than the "single-task" and attention model which use ingredient-only attributes. With additional attributes, "single-task (BF)" manages to distinguish ingredients undergone drastic appearance changes because of cutting and cooking methods. Nevertheless, due to the lack of training examples, the performance of "single-task (BF)" is not as good as our proposed architecture. For example, the ingredient "wild rice stem" has limited examples and is being applied to different cutting methods. In such circumstance, "single-task (BF)" will perform poorly as compared to our model. On the other hand, when sufficient training examples are available, for example, different type of "egg" that are cut and cooked under various ways, "single-task (BF)" indeed exhibits better performance.

4.3.5 Response map

A by-product of our multi-task model is the capability of locating ingredients. We visualize the result in a response map, which is formed by converting the prediction

Table 4.5: Comparison of different deep architectures. The best performance is highlighted in bold font.

	Method	MRR	R@1	R@5	R@10	R@20
FC	Single-task	0.070	0.065	0.188	0.268	0.374
	Single-task (BF)	0.113	0.107	0.284	0.393	0.521
Region	Single-task	0.098	0.082	0.229	0.329	0.487
	Single-task (BF)	0.139	0.188	0.312	0.453	0.640
	Attention model [23]	0.136	0.081	0.293	0.503	0.707
	Ours	**0.153**	**0.204**	**0.338**	**0.518**	**0.742**



(a) query (b) shrimp (c) cucumber (d) tomato

Figure 4.7: Ingredient localization: (a) input image; (b)–(d) response maps of ingredients.

score of an ingredient on an image grid into pixel intensity value. Figure 4.7 shows the response maps of three ingredients for a query image. Generally speaking, the better the result of localization is, the higher the prediction accuracy. Figure 4.8 visualizes more results for different kinds of ingredient composition. For dishes with well-separated ingredients such as Figure 4.8(a), our model often achieves high prediction as well as localization accuracy despite different ingredients being cooked or cut with different methods. Localization becomes challenging when ingredients are mixed as in Figure 4.8(b), but our model still manages to show reasonable result. As our model considers only region-level information, the ingredient labels at different locations of dishes could be inconsistent. For example in Figure 4.8(c), "garlic sprout" is sometimes predicted as "green onion" which has similar visual appearance. Similarly for the ingredients "lamp" and "pork". Some

of the consistency can indeed be removed by noise filtering through techniques such as region merging and common sense rules that certain kinds of meats are seldom cooked together to reduce the risk of bacteria cross-contamination. Finally, although multi-dish segmentation is not considered, our model is able to locate ingredients for different dishes in a picture as shown in Figure 4.8(d).

4.4 Summary

We have presented a multi-task deep learning architecture for addressing the challenge of recognizing ingredients under different cutting and cooking methods. Particularly, we shed light that, instead of coupling all three attributes to generate exponential number of ingredient labels for model training, learning the attributes in multi-task manner can generate predictions feasible for recipe retrieval. The model suffers less from the need of a large amount of learning samples and is easier to be trained with a smaller number of network parameters. Experimental results basically confirm the merit of using cutting and cooking attributes in recognizing diverse appearance of ingredients. More importantly, leveraging three attributes altogether enables a more effective way of ranking recipes that share the same or similar set of ingredients. Despite these encouraging results, ingredients with similar visual outlook and ingredients that are covered under sauces or being occluded remain difficult to be identified, which affect the retrieval effectiveness.

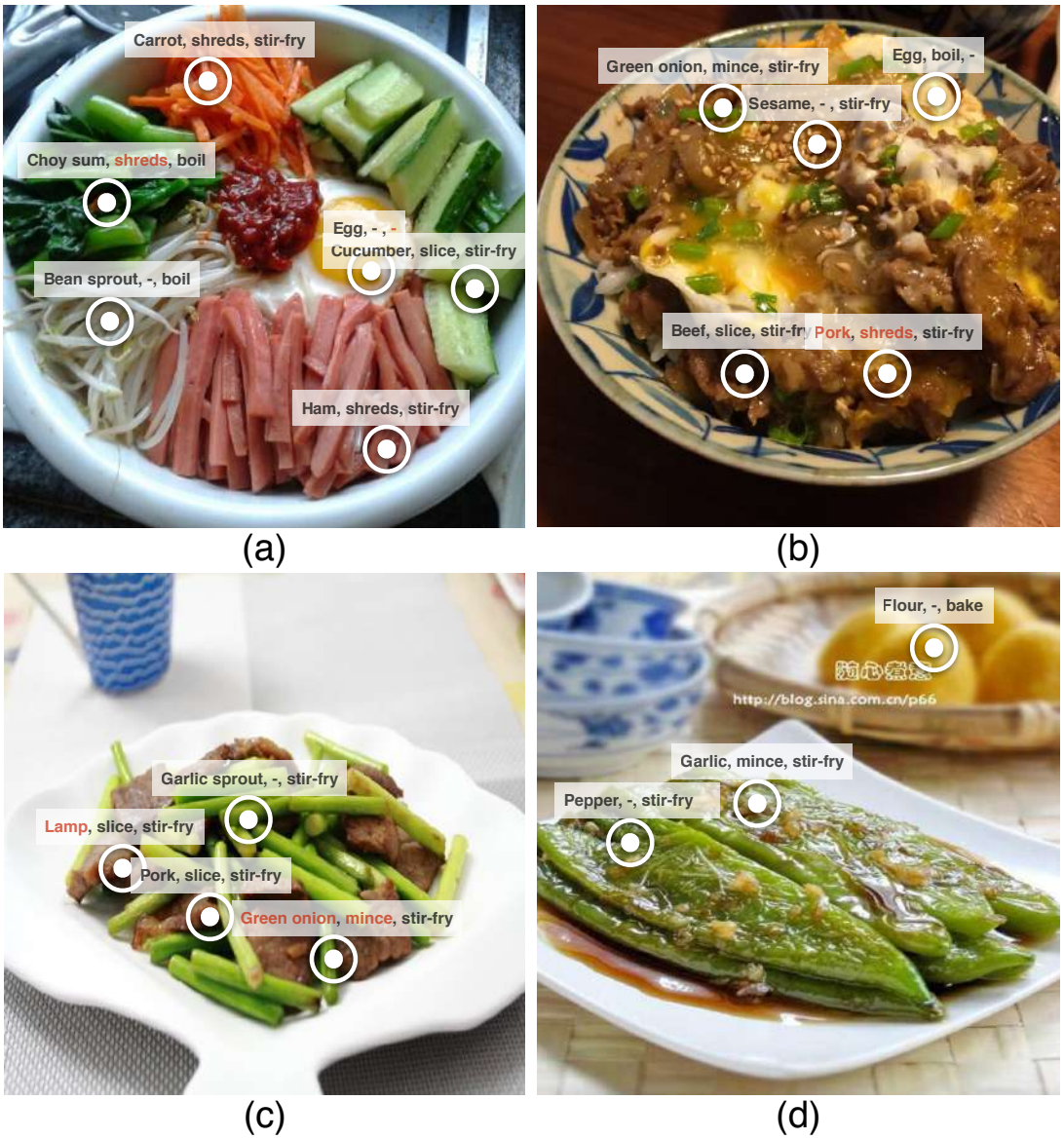


Figure 4.8: Examples of attribute prediction and localization. The circle indicates the most confident location of an ingredient. False positives are marked in red.

CHAPTER 5

CROSS MODEL RETRIEVAL WITH STACKED ATTENTION MODEL

In the previous chapters, we introduced ingredient recognition as well as rich food attribute recognition based recipe retrieval methods. Although the recognition based methods achieve quite promising recipe retrieval performance even for queries from unknown food categories, they require explicit labeling of attributes, which is time-consuming. Therefore, in this chapter, we study the problem of cross-modal recipe retrieval from the perspective of cross model learning.

This Chapter explores the recent advances in cross-modality learning for recipe retrieval. Specifically, given food pictures and their associated recipes, our aim is to learn a model that captures their correspondence by learning a joint embedding space for visual-and-text translation. We exploit and revise a deep model, stacked attention network (SAN) [1], originally proposed for visual question-answering for our purpose. The model learns the correspondence through assigning heavier weights to the attended regions relevant to the ingredients extracted from recipes. Notice that ingredient-irrelevant but context-relevant regions may also be useful for recipe retrieval, for example, the "casserole" regions would be effective in identifying dishes like "casserole rice noodles" or "casserole tofu". A similar case also happens for ingredients (e.g., water) that appear in a dish but are not written in the recipe. Therefore, directly ignoring these regions would decrease retrieval performance. Thanks to attention mechanism, our model will not completely ignore context-relevant regions but assign weights that are usually lower than ingredient regions, so long as the contextual information is useful in reducing training error.

For the task of recipe retrieval, fortunately the learning does not require much effort in labeling training examples. There are already millions of food-recipe pairs uploaded by professional and amateur chefs on various cooking websites, which can be freely leveraged for training. We demonstrate that using these online resources, a fairly decent model can be trained for recipe retrieval with minimal labeling effort. As input to SAN includes ingredients, the model has higher generalization ability in recognizing food categories unseen during training, as long as all or most ingredients are known. Furthermore, as ingredient composition is considered in SAN, the chance of retrieving the best-match recipes is also enhanced. To this end, the contribution of this Chapter lies in addressing of food recognition as a recipe retrieval problem. Under this umbrella, the problem is turned into cross-modality feature learning, which can integrally model three inter-related problems: scalable food recognition, fine-grained ingredient recognition and best-match recipe retrieval.

The rest of this Chapter is organized as follows: Section 5.1 introduce the stacked attention network, while Section 5.2 presents the experimental results for recipe retrieval. Finally, Section 5.3 summarizes this chapter.

5.1 Stacked Attention Network (SAN)

Figure 5.1 illustrates the SAN model, with visual and text features respectively extracted from image and recipe as input. The model learns a joint space that boosts the similarity between images and their corresponding recipes. Different from [1], where the output layer is for classification, we modify SAN so as to maximize the similarity for image-recipe pairs.

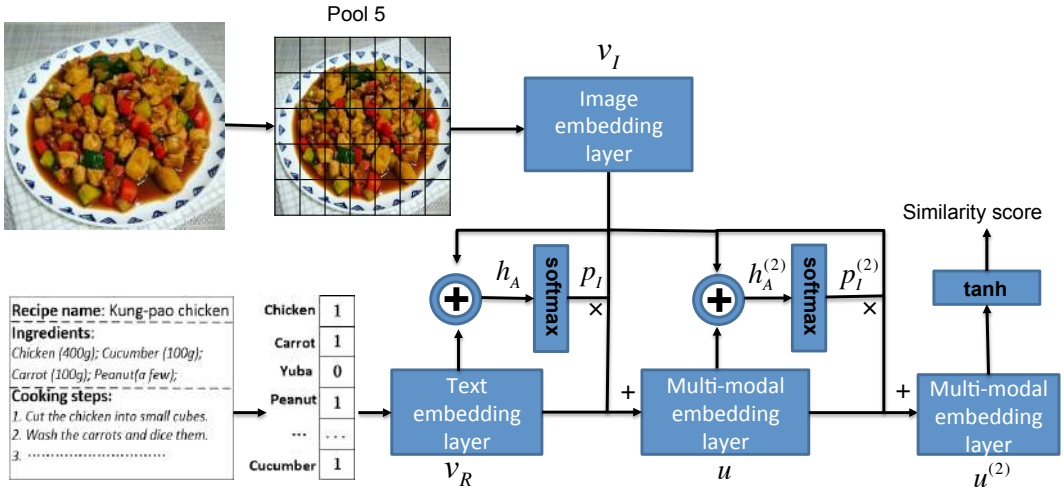


Figure 5.1: SAN model inspired from [1] for joint visual-text space learning and attention localization.

5.1.1 Image Embedding Feature

The input visual feature is the last pooling layer of DCNN – Pool5 – that retains the spatial information of the original image. The dimension of Pool5 feature is 512×14×14, corresponding to 14×14 or 196 spatial grids of an image. Each grid is represented as a vector of 512 dimensions. Let's denote $\boldsymbol{F}_I$ as the Pool5 feature and is composed of regions f_i, $i \in [0,195]$. Each region f_i is transformed to a new vector or embedding feature as follows:

$$\boldsymbol{V}_I = \tanh(\boldsymbol{W}_I \boldsymbol{F}_I + \boldsymbol{b}_I), \tag{5.1}$$

where $\boldsymbol{V}_I \in \mathbb{R}^{d \times m}$ is the transformed feature matrix, with d as the dimension of the new vector and $m = 196$ is the number of grids or regions. The embedding feature of f_i is indexed by the i-th column of $\boldsymbol{V}_I$, denoted as $\boldsymbol{v}_i$. The transformation is performed region-wise, $\boldsymbol{W}_I \in \mathbb{R}^{d \times 512}$ is the transformation matrix and $\boldsymbol{b}_I \in \mathbb{R}^d$ is the bias term.

5.1.2 Recipe Embedding Feature

A recipe is represented as a binary vector of ingredients, denoted as $\boldsymbol{r} \in \mathbb{R}^t$. The dimension of the vector is t corresponding to the size of the ingredient vocabulary. Each entry in $\boldsymbol{r}$ indicates the presence (1) or absence (0) of a particular ingredient in a recipe. As Pool5 feature, the vector is embedded into a new space as follows:

$$\boldsymbol{v}_R = \tanh(\boldsymbol{W}_R \boldsymbol{r} + \boldsymbol{b}_R), \tag{5.2}$$

where $\boldsymbol{W}_R \in \mathbb{R}^{d \times t}$ is the embedding matrix and $\boldsymbol{b}_R \in \mathbb{R}^d$ is the bias vector. Note that, for joint learning, the embedding features of recipe ($\boldsymbol{v}_R \in \mathbb{R}^d$) and Pool5 region ($i$-th column of $\boldsymbol{V}_I$) have the same dimension.

5.1.3 Joint embedding feature

The attention layer is to learn the joint feature by trying to locate the visual food regions that correspond to ingredients. There are two transformation matrices, $\boldsymbol{W}_{I,A} \in \mathbb{R}^{k \times d}$ for image I and $\boldsymbol{W}_{R,A} \in \mathbb{R}^{k \times d}$ for recipe R, mimicking the attention localization, formulated as follows:

$$\boldsymbol{H}_A = \tanh(\boldsymbol{W}_{I,A} \boldsymbol{V}_I \oplus (\boldsymbol{W}_{R,A} \boldsymbol{v}_R + \boldsymbol{b}_A)), \tag{5.3}$$

$$\boldsymbol{p}_I = \mathrm{softmax}(\boldsymbol{W}_P \boldsymbol{H}_A + \boldsymbol{b}_P), \tag{5.4}$$

where $\boldsymbol{H}_A \in \mathbb{R}^{k \times m}$, $\boldsymbol{p}_I \in \mathbb{R}^m$, $\boldsymbol{W}_P \in \mathbb{R}^{1 \times k}$. We denote by $\oplus$ the addition of a matrix and a vector performed by adding each column of the matrix by the vector. Note that $\boldsymbol{p}_I$ aims to capture the attention, or more precisely relevance, of image regions to a recipe. The significance of a region f_i is indicated by the value in the corresponding element $p_i \in \boldsymbol{p}_I$.

The joint visual-text feature is basically generated by adding the embedding features $\boldsymbol{V}_I$ and $\boldsymbol{v}_R$. To incorporate attention value, regions $\boldsymbol{v}_i$ are linearly weighted

and summed (equation-5.5) before the addition operation with $\boldsymbol{v}_R$ (equation-5.6), as follows:

$$\tilde{\boldsymbol{v}_I} = \sum_{i=1}^{m} p_i \boldsymbol{v}_i, \tag{5.5}$$

$$\boldsymbol{u} = \tilde{\boldsymbol{v}_I} + \boldsymbol{v}_R, \tag{5.6}$$

where $\tilde{\boldsymbol{v}_I} \in \mathbb{R}^d$, and $\boldsymbol{u} \in \mathbb{R}^d$ represent the joint embedding feature.

As suggested in [1], progressive learning by stacking multiple attention layers can boost the performance, but will heavily increase the training cost. We consider two-layer SAN, by feeding the output of the first attention layer, $\boldsymbol{u}^{(1)}$, into the second layer to generate a new joint embedding feature $\boldsymbol{u}^{(2)}$ as follows:

$$\boldsymbol{H}_A^{(2)} = \tanh(\boldsymbol{W}_{I,A}^{(2)} \boldsymbol{V}_I \oplus (\boldsymbol{W}_{R,A}^{(2)} \boldsymbol{u} + \boldsymbol{b}_A^{(2)})), \tag{5.7}$$

$$\boldsymbol{p}_I^{(2)} = \text{softmax}(\boldsymbol{W}_P^{(2)} \boldsymbol{H}_A^{(2)} + \boldsymbol{b}_P^{(2)}), \tag{5.8}$$

$$\tilde{\boldsymbol{v}_I^{(2)}} = \sum_i p_i^{(2)} \boldsymbol{v}_i, \tag{5.9}$$

$$\boldsymbol{u}^{(2)} = \tilde{\boldsymbol{v}_I^{(2)}} + \boldsymbol{u}. \tag{5.10}$$

As $\boldsymbol{p}_I^{(2)}$ indicates the region relevancy, the attention map can be visualized by back projecting the attention value p_i to its corresponding region f_i, followed by upsampling to the original image size with bicubic interpolation.

5.1.4 Objective Function

To this end, the similarity between food image and recipe is generated as follows:

$$S\langle \boldsymbol{V}_I, \boldsymbol{v}_R \rangle = \tanh(\boldsymbol{W}_{u,s} \boldsymbol{u}^{(2)} + b_s), \tag{5.11}$$

where $\boldsymbol{W}_{u,s} \in \mathbb{R}^{1 \times d}$ and $b_s \in \mathbb{R}$ is the bias. $S\langle \boldsymbol{V}_I, \boldsymbol{v}_R \rangle$ outputs a score indicating the association between the embedding features of image and recipe. The learning

is based on the following rank-based loss function with a large margin form as the objective function:

$$\mathcal{L}(\boldsymbol{W}, D_{trn}) = \sum_{(\boldsymbol{V}_I, \boldsymbol{v}_R^+, \boldsymbol{v}_R^-) \in D_{trn}} \max(0, \triangle + S\langle \boldsymbol{V}_I, \boldsymbol{v}_R^- \rangle - S\langle \boldsymbol{V}_I, \boldsymbol{v}_R^+ \rangle). \tag{5.12}$$

The training set, D_{trn}, consists of triplets in the form of $(\boldsymbol{V}_I, \boldsymbol{v}_R^+, \boldsymbol{v}_R^-)$, where $\boldsymbol{v}_R^+$ ($\boldsymbol{v}_R^-$) is true (false) recipe for food V_I. The matrix $\boldsymbol{W}$ represents the network parameters, and $\triangle \in (0, 1)$ controls the margin in training and is cross-validated.

5.2 Experiments

5.2.1 Settings and Evaluation

Here we detail the parameter setting of SAN. The dimension of the embedding feature is set to $d = 500$ for both Pool5 regional and recipe features, while the dimension of h_A is $k = 1,024$ for equations 3 and 7. Through cross-validation, the hyperparameter $\triangle$ for the loss function is set as 0.2. SAN is trained using stochastic gradient descent with momentum set as 0.9 and the initial learning rate as 1. The size of mini-batch is 50 and the training stops after 10 epochs. To prevent overfitting, dropout [76] is used. The Pool5 feature can be extracted from any DCNN model. We employ the multi-task VGG proposed in Chapter 3, which reported the best performances on two large food datasets, VIREO Food-172 [21] and UEC Food-100 [77]. The model, as shown in Figure 5.2, has two pathways, one for classifying 172 food categories while another for labeling 353 ingredients. For a fair comparison, all the compared approaches in the experiment are using multi-task VGG features, either Pool5 or deep ingredient feature (fc7), as shown in Figure 5.2.

As the task is to find the best possible recipe given a food picture, the following two measures are employed for performance evaluation:

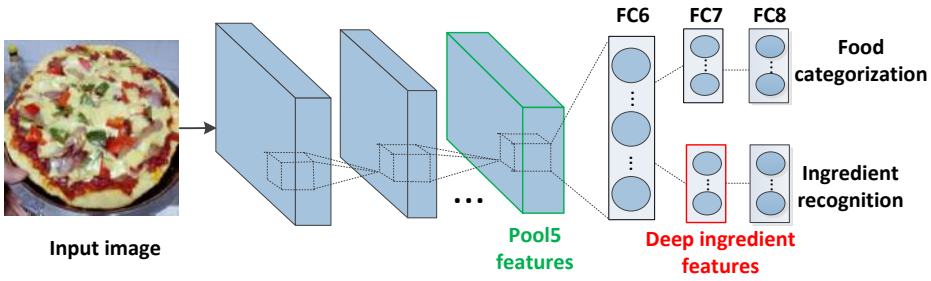


Figure 5.2: Multi-task VGG model in Chapter 3 offering Pool5 and deep ingredient features for cross-modal joint space learning.

- Mean reciprocal rank (MRR): MRR measures the reciprocal of rank position where the ground truth recipe is returned, averaged over all the queries. This measure assesses the ability of the system to return the correct recipe at the top of the ranking. The value of MRR is within the range of [0, 1]. A higher score indicates a better performance.

- Recall at Top-K (R@K): R@K computes the fraction of times that a correct recipe is found within the top-K retrieved candidates. R@K provides an intuitive sense of how quickly the best recipe can be located by investigating a subset of the retrieved items. As MRR, a higher score also indicates a better performance.

5.2.2 Dataset

The dataset is composed of 61,139 image-recipe pairs crawled from the "Go Cooking"[1] websites. Each pair consists of a recipe and a picture of resolution 448×448. The dataset covers different kinds of food, like Chinese dishes, snacks, dessert, cookies and Chinese-style western food, as shown in Figure 5.3. Each recipe includes the list of ingredients and cooking procedure. As the recipes were uploaded

[1]https://www.xiachufang.com/

Figure 5.3: Examples of dishes in the dataset.

by amateurs, the naming of ingredients is not always consistent. For example, "carrot" is sometimes called as "carotte". We manually rectified the inconsistency and compiled a list of 5,990 ingredients, both visible and non-visible (e.g., "honey"), from these recipes. The list, represented as a binary vector indicating the presence or absence of particular ingredients in a recipe, serves as input to the SAN model. Note that in some cases the cooking and cutting methods are directly embedded into the name of ingredient, for example, "tofu" and "tofu piece", "egg" and "steamed egg". The dataset is split into three sets: 54,139 pairs for training, 2,000 pairs for cross-validation, and 5,000 pairs for testing. Furthermore, we selected 1,000 images from the testing set as queries to search against the 5,000 recipes. The queries are sampled in such a way that there are around 45% of them (446 queries) belonging to food categories unknown to SAN and multi-task VGG models. In addition, around 85% of the queries have more than one relevant recipe. We recruit a homemaker, who has cooking experience, to manually pick the relevant recipes for each of the 1,000 queries. The homemaker is instructed to label relevant recipes based on title similarity in recipes, titles that are named differently because of geographic regions or sharing almost the same cooking pro-

cedure with similar key ingredients. For example, the dish "sauteed tofu in hot and spicy sauce" is sometimes called as "mapo tofu" in the restaurant menu. In the extreme case, some queries have more than 60 relevant recipes. On average each query has 9 number of relevant recipes. Note that the testing queries are designed in these ways so as to verify the two major claims in this chapter, i.e., the degree in which the learnt model can generalize to unseen food categories (Section 4.4) and the capability in finding the best-matched recipe (Section 4.5).

5.2.3 Performance Comparison

We compared SAN to both shallow and deep models for cross-modal retrieval as following. The inputs to these models are the deep ingredient feature (fc7) of the multi-task VGG model and the ingredient vector of 5,990 dimensions. The Pool5 feature is not used due to its high dimensionality (14×14×512). As reported in [78], simply concatenating the features from 14×14 grids performs worse than fc7 in visual recognition.

- Canonical Correlation Analysis (CCA)[55]: CCA is a classic way of learning latent subspace between two views or features by maximizing the correlation between them. Two linear mapping functions are learnt for projecting features into subspace.

- Partial Least Squares (PLS)[56]: Similar to CCA, PLS learns two linear mapping functions between two views. Instead of using cosine similarity as in CCA, PLS uses dot product as the function for measuring correlation.

- DeViSE [52]: DeViSE is a deep model with two pathways which respectively learn the embedded features of recipe-image pairs to maximize their similarities. Note that, instead of directly using word2vec as in [52], the embedded

feature of ingredients is learnt from the training set of our dataset. This is simply because word2vec is learnt from documents such as news corpus [79] and lacks specificity in capturing information peculiar to ingredients. Different from SAN, DeViSE is not designed for attention region localization.

- DeViSE++: We purposely include a variant of DeViSE, which takes the hand-cropped regions of food as input to the deep model. The cropping highlights the target food region and basically removes the background or irrelevant part of food pictures. The aim of using DeViSE++ is to gate the potential improvement over DeViSE when only food region is considered, and more importantly, to justify the merit of SAN in identifying appropriate attention region in comparison to the hand-cropped region.

- Multi task [21]: In Multi task [21] model, ingredient recognition is formulated as a problem of multi-task learning and the learnt semantic labels as well as the external knowledge of the contextual relations among ingredients are utilized for recipe retrieval.

Table 5.1: MRR and R@K for recipe retrieval. The best performance is highlighted in bold font.

Method	MRR	R@1	R@5	R@10	R@20	R@40	R@60	R@80	R@100
CCA	0.055	0.023	0.079	0.123	0.182	0.262	0.329	0.371	0.413
PLS	0.032	0.009	0.039	0.073	0.129	0.219	0.284	0.338	0.398
DeViSE	0.049	0.016	0.060	0.108	0.182	0.300	0.391	0.456	0.524
DeViSE++	0.050	0.016	0.059	0.105	0.174	0.307	0.404	0.471	0.531
Multi task [21]	0.097	**0.051**	0.128	0.184	0.251	0.324	0.372	0.408	0.438
SAN	**0.115**	0.048	**0.161**	**0.249**	**0.364**	**0.508**	**0.601**	**0.671**	**0.730**

Table 5.1 lists the results of different approaches. Deep models basically outperform shallow models in terms of recall at the depth of 20 and beyond. In contrast to PLS, which does not perform score normalization, CCA manages to outperform DeViSE in terms of MRR and R@K for $K < 20$. Among all these

approaches, the proposed model SAN consistently exhibits the best performance in terms of MRR. Compared to DeViSE and Multi task, SAN achieves a relative improvement of 130% and 18% in MRR, respectively. In terms of R@K, SAN performs significantly better than DeViSE and doubles its performance at R@20, which is fairly impressive. Compared with Multi task model, SAN also performs much better when $K > 5$, and the performance gap becomes larger when the depth increases.

To further provide insights, Figure 5.4 visualizes the attention maps learnt from SAN while comparing to Pool5 feature maps. From the figure, it is obvious that the learnt attention model can locate the ingredient regions more accurately than Pool5 feature maps.

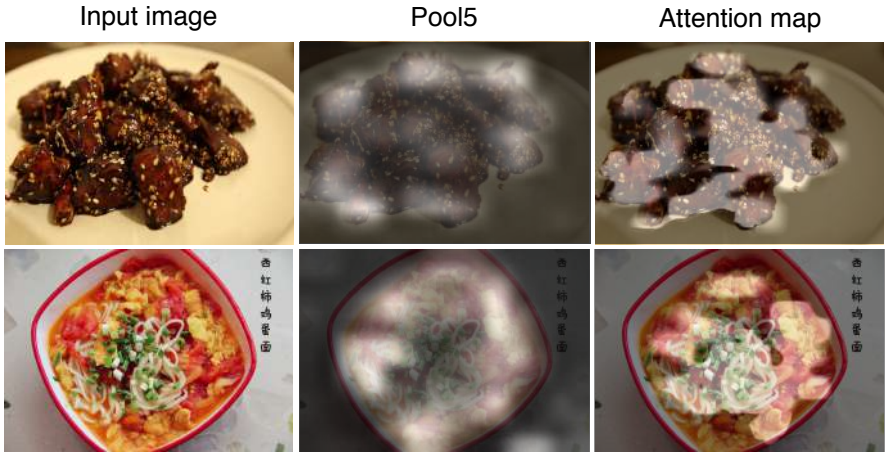


Figure 5.4: Visualizing attention maps, the learnt attention regions are highlighted in white.

To observe the difference between the two learnt attention layers, we visualize the attention maps $\boldsymbol{p}_I$ and $\boldsymbol{p}_I^{(2)}$. Four examples are shown in Figure 5.5. From the figure, we can observe that the second attention layer reduces the noises in the first layer and hence is more accurate in localizing ingredient regions.

Despite the encouraging performance by SAN, the value of R@1 is only around

Figure 5.5: Visualization of two attention layers.

0.05. Figure 5.6 shows some successful and near-miss examples. The first two pictures show query images where all visible ingredients are clearly seen. SAN manages to retrieve the ground-truth recipe at top-1 rank in such cases. In the third example, SAN ranks "grilled salmon" higher than "fried salmon" as the current model does not consider cooking attributes. In addition, SAN overlooks the beef and peanuts which are mixed and partially occluded by salmon, while confused by the ingredients of similar appearance, i.e., caviar and red pepper, bean sprout and basil. The last query image shows an example of how non-visible ingredients, flour in this example, affect the ranking. The flour is used to make the dish into round shape, and this knowledge does not seem to be learnt by SAN.

Another result worth noticing is that there is no performance difference between DeViSE and DeViSE++. While DeViSE is not designed for attention localization, the model seems to have the ability to exclude irrelevant background regions from recognition. To provide further insights, Figure 5.7 shows some examples visu-

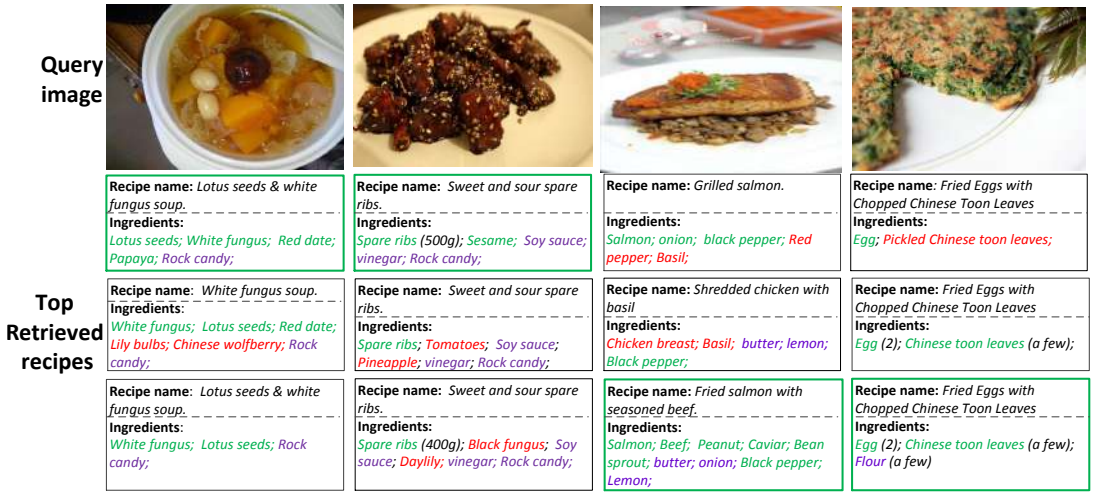


Figure 5.6: Examples of top-3 retrieved recipes (ranked from top to bottom). Ground-truth recipe is marked in green. The ingredients in different colours have different meanings: green – true positive, purple – true positive but non-visible in dish, red – false positive.

alizing the attention regions highlighted by SAN and in contrast to hand-crafted regions. In the first example, the region attended by SAN is about the same as the region manually cropped. In this case, DeViSE+ and SAN use to have similar performance. The next two examples highlight the superiority of SAN in excluding soup and foil as attention regions, which cannot be easily done by simple region cropping. SAN significantly outperforms DeViSE in such examples. Finally, the last example shows a typical case that SAN only highlights part of dishes as attention. While there is no direct explanation of why certain food regions are ignored by SAN for joint space learning, it seems that SAN has the ability to exclude regions that are vague and hard to be recognized even by human.

5.2.4 Finding the best matches recipes

Recall that around 85% of query images have more than one relevant recipe. This section examines the ability of SAN in identifying the best (or ground-truth) recipe

Figure 5.7: (a) Examples contrasting the manually cropped region (green bounding box), (b) the learnt attention region (masked in white) by SAN.

from the testing set composed of 5,000 recipes. Figure 5.8 shows the performance of the best match recipe retrieval comparing with relevant recipe retrieval. For recall@top5, the performance of relevant recipe retrieval improves when the number of relevant recipe increases while the trend is opposite for best-match recipe retrieval. To provide insights, we select the queries that retrieve at least one relevant

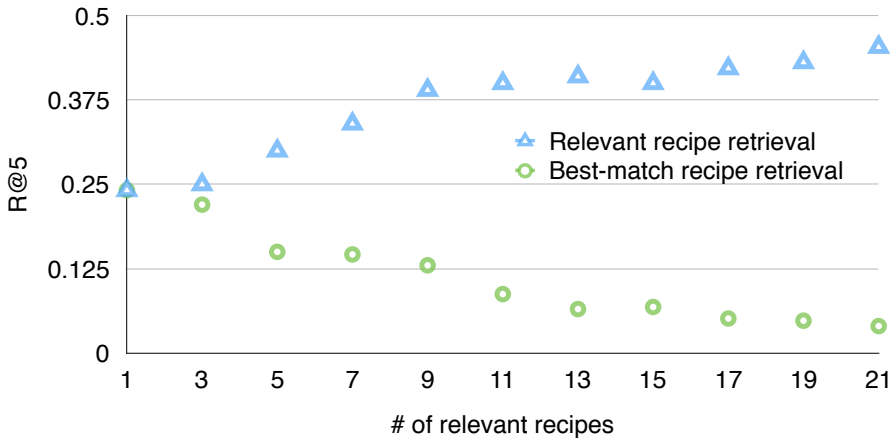


Figure 5.8: Performance of best match recipe retrieval and relevant recipe retrieval.

recipe (excluding ground-truth recipe) within the top-5 position for analysis. The purpose is to show how the performance of best-match recipe retrieval is impacted by the increasing number of relevant recipes. We divide the selected queries into

Table 5.2: Performance comparison between SAN and DeViSE in retrieving best–match recipes.

		R@1		R@5		R@10	
Recipe #	Query #	SAN	DeViSE	SAN	DeViSE	SAN	DeViSE
2-3	33	0.21	0.15	0.67	0.48	0.82	0.76
4-7	66	0.18	0.17	0.56	0.53	0.70	0.67
8-11	54	0.17	0.15	0.54	0.30	0.60	0.50
11-15	38	0.13	0.08	0.47	0.39	0.63	0.55
16-30	48	0.06	0.06	0.46	0.39	0.62	0.52
31-61	25	0.08	0.08	0.28	0.26	0.44	0.44

seven groups with the intention to make the number of queries in each group as even as possible. Note that, as the numbers of recipes distribute in a long-tail like manner, the recipe numbers in each group are uneven. Table 5.2 lists the performance for each group. As can be seen from the table, the difficulty of finding best-match is proportional to the number of relevant recipes. Compared to DeViSE, SAN generally shows better performance for R@1. As the number of recipes increases, they tie in performance. Nevertheless, while looking deeper into the list, SAN consistently outperforms DeViSE in terms of R@5 and R@10. Two main reasons that ground truth recipes are not ranked higher are due to occluded ingredients and the use of different non-visible ingredients. Two such examples are shown in the last two pictures of Figure 5.6.

5.2.5 Generalization to unknown categories

Figure 5.9 further shows the performance of SAN to unseen categories. As expected, the performance is not as good as that for the food categories known to SAN and multi-task VGG. Figure 5.10 shows both success and failure examples of recipe retrieval. Basically, when the ingredients of unknown food categories are previously seen and can be correctly identified, SAN performs satisfactorily.

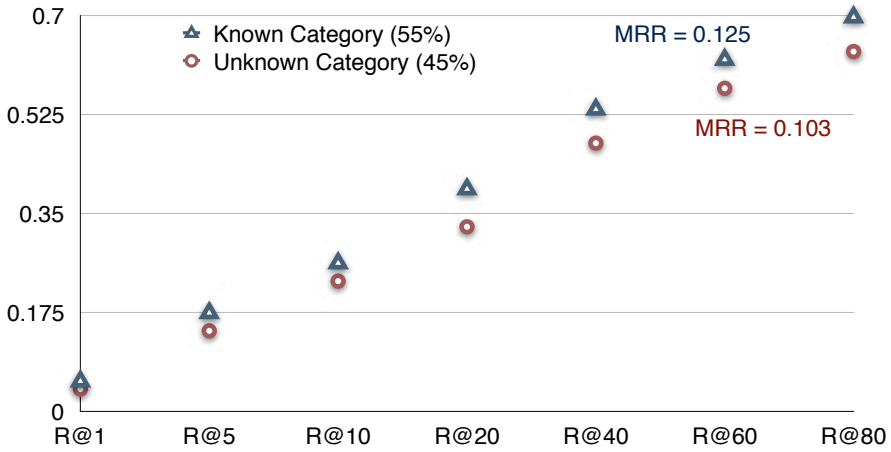


Figure 5.9: Generalization of SAN to unseen food categories.

In contrast, when some ingredients, especially key ingredients, are unknown, the model will likely fail in retrieving relevant recipes. In the first example, the ingredients are correctly recognized despite that the dish belongs to unseen food categories. As results, our model is able to rank the best match recipe at the top-1 place. However, when the ingredient is covered by flour (second example), the model is unlikely to recognize the ingredients and hence fails to retrieve the correct recipes at top ranks. Finally, when the dish contains unseen key ingredients, for example, "fishwort" in the third example, our model will fail.

We further compare the generalization ability of our model with DeViSE and Multi task [21]. The retrieval performances are evaluated on 446 queries that come from unknown food categories. As can be seen from the Figure 5.11, our model enjoys higher generalization ability and the performance gap becomes larger when the depth of recall increase. The better generalization ability of our model verifies the advantages of cross-modal learning on region-level with stacked attention networks.

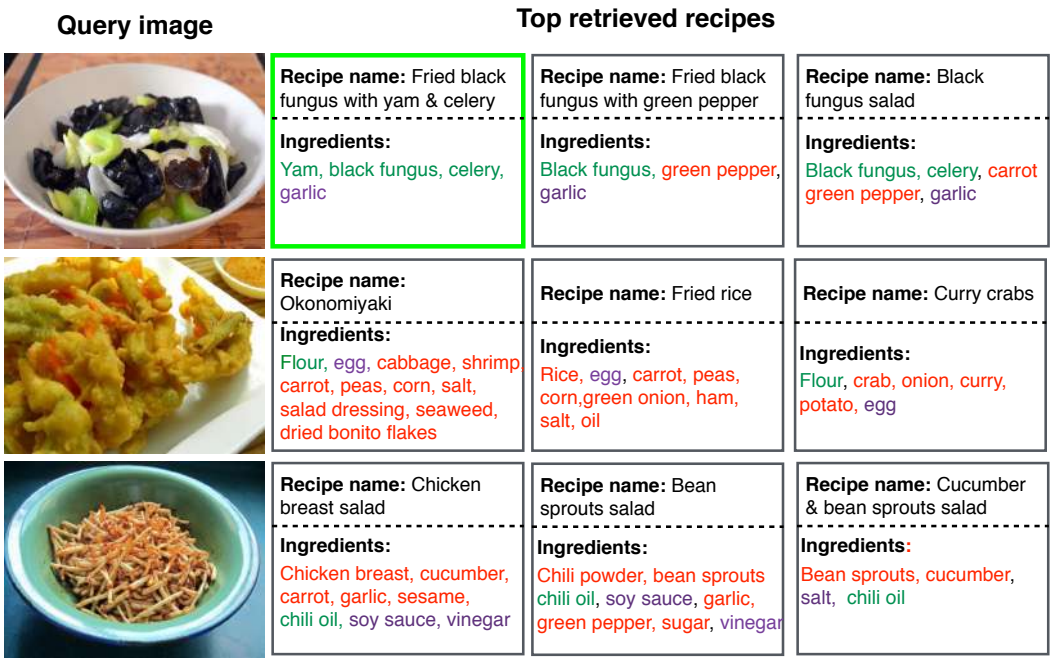


Figure 5.10: Examples of top-3 retrieved recipes for unknown food categories. Ground-truth recipe is marked in green. The ingredients in different colours have different meanings: green – true positive, purple – true positive but non-visible in dish, red – false positive.

5.3 Summary

We have presented a deep model for learning the commonality between image and text at the fine-grained ingredient level. The power of model comes from the ability to infer attended regions relevant to the ingredients extracted from recipes. This peculiarity enables retrieval of best-match recipes even for unseen food category. The merit of our approach is that it requires much less labeling efforts compared to learning individual ingredient classifiers. The experimental results basically verify our claims that the model can deal with unknown food categories to the extent that at least key ingredients are seen during training. In addition, SAN exhibits consistently better performance than DeViSE, showing the advantage of fine-grained ingredient analysis at the regional level for best-match recipe retrieval.

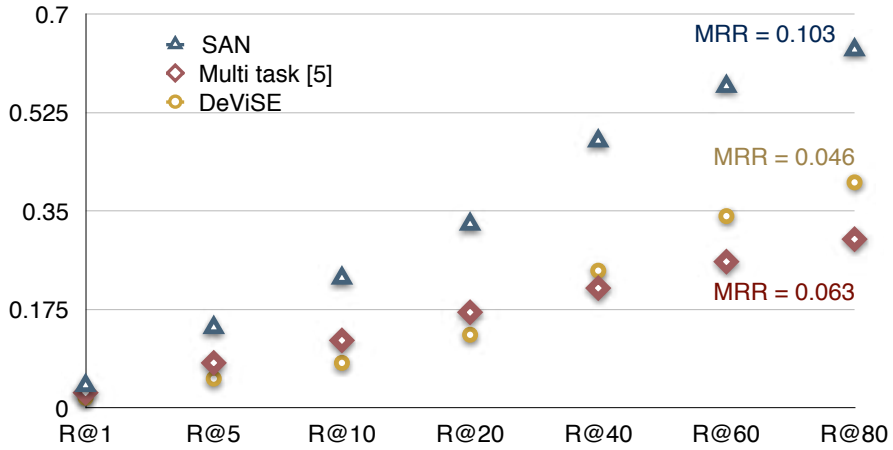


Figure 5.11: Comparison of generalization ability among different methods.

While the current model does not consider food category information, it is expected that such information will boost performance especially when there are errors in ingredient localization and attention modeling. How to incorporate food category information into the current model is worth further investigation. Finally, our current model can be extended to explicitly model cutting and cooking attributes in cross-modal learning, which could address some limitations identified in the experiments. In addition, as the attention layers couple both visual and text features, the embedding features cannot be offline indexed and have to be generated on-the-fly when the query image is given. This poses the limitation on retrieval speed for online application, which is an issue needs to be further researched.

CHAPTER 6

DEEP UNDERSTANDING OF COOKING PROCEDURE

In Chapter 5, we presented a stacked attention model that learns the joint space between ingredient regions of food images and ingredient list extracted from text recipes. Cooking procedure, which contains rich cooking information including how ingredients are cut and cooked, has been ignored by the previous chapter. However, as discussed in Chapter 4, using ingredient alone cannot deal with the situation when the same set of ingredients constitute different dishes. Therefore, in this Chapter, we take into account the cooking procedure for the joint space learning.

Finding a right recipe that describes the cooking procedure for a dish from just one picture is inherently a difficult problem. Food preparation undergoes a complex process involving raw ingredients, utensils, cutting and cooking operations. This process gives clues to the multimedia presentation of a dish (e.g., taste, colour, shape). However, the description of the process is implicit, implying only the *cause* of dish presentation rather than the visual *effect* that can be vividly observed on a picture. Therefore, different from other cross-modal retrieval problems in the literature, recipe search requires the understanding of a textually described procedure to predict its possible consequence on visual appearance. In this chapter, we approach this problem from the perspective of attention modeling. Specifically, we model the attentions of words and sentences in a recipe and align them with its image feature such that both text and visual features share high similarity in multi-dimensional space. The novelty of our work originates from the proposal

of an attention mechanism for this problem. Despite technically straightforward, this is the first attempt in literature that investigates the extent which attention can deal with the causality effect while being able to demonstrate impressive performance on cross-modal recipe retrieval. In addition, we provide a unified way of dealing with three sections of information (i.e., title, ingredient, instruction) in a recipe.

This chapter is organized as follows. Section 6.1 presents the basic framework of our proposed attention mechanism, while Section 6.2 presents the experimental results for cross-modal recipe retrieval. Finally, Section 6.3 summarizes this chapter.

6.1 Methodology

Figure 6.1 depicts the basic framework of our proposed attention mechanism. First, different modalities are input to both ends of the deep model for representation learning. Recipes, in particular, are split into three sections (title, ingredient, instruction) based on different levels of information granularities. These sections are encoded separately by attention mechanism into three representations, which are eventually concatenated as a recipe representation Figure 6.1(a). Together with image representation which is learnt through convolutional network Figure 6.1(b), the proposed model learns to maximize the cosine similarity between textual recipes and their associated food images. The similarity learning is carried out through two representation transformations that aim to make recipe and image features as alike as possible Figure 6.1(c).

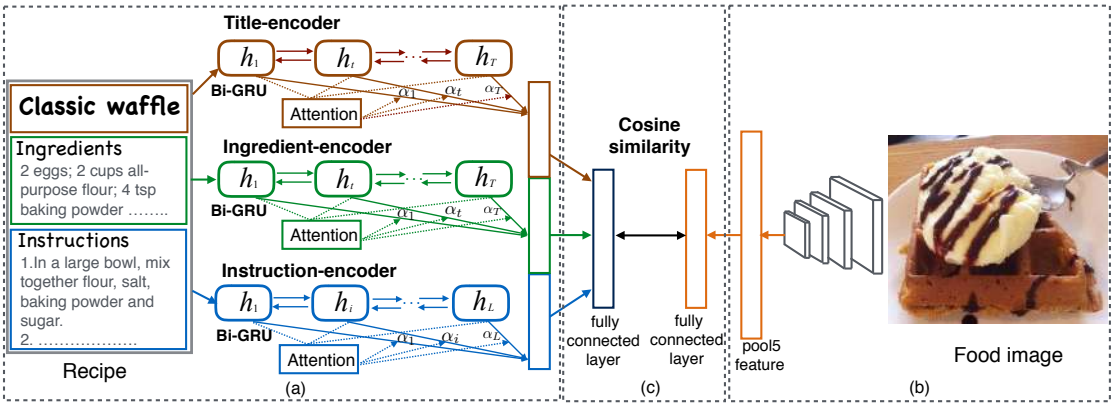


Figure 6.1: Framework overview: (a) recipe representation learning; (b) image feature learning; (c) joint-embedding space learning.

6.1.1 Recipe representation

Title encoder. Each recipe has a title as the name of dish. As expected, the title uses to elicit dish peculiarity by capturing food uniqueness directly into name. The characterization of food uniqueness is multi-perspective in nature, ranging from taste, style (e.g., "old fashion", "nouvella", "home-made"), cuisine and geography region, ingredient and cooking method, to even cooking utensil. Examples include "peek potato and bacon casserole recipe", "caramelized beef skewers" and "home-made healthy granola bars". For title representation, the aim of attention model is to assign higher weights to words that directly link to food content relative to contextually relevant terms about style and location.

Given a title with words $\boldsymbol{w}_t$, $t \in [0, T]$, we first embed individual word to a vector through a matrix $\boldsymbol{W}_e$, $\boldsymbol{x}_t = \boldsymbol{W}_e \boldsymbol{w}_t$. The title is treated as a sequence and a bidirectional gated recurrent unit (GRU) [80] is employed to encode the word sequence. The bidirectional GRU is composed of a forward $\overrightarrow{GRU}$ which reads title

from $\boldsymbol{w}_1$ to $\boldsymbol{w}_T$ and a backward $\overleftarrow{GRU}$ which reads from $\boldsymbol{w}_T$ to $\boldsymbol{w}_1$, defined as

$$\boldsymbol{x}_t = \boldsymbol{W}_e \boldsymbol{w}_t, t \in [1, T], \tag{6.1}$$

$$\overrightarrow{\boldsymbol{h}_t} = \overrightarrow{GRU}(\boldsymbol{x}_t), t \in [1, T], \tag{6.2}$$

$$\overleftarrow{\boldsymbol{h}_t} = \overleftarrow{GRU}(\boldsymbol{x}_t), t \in [1, T]. \tag{6.3}$$

The representation of a word $\boldsymbol{w}_t$ can by obtained by concatenating the forward hidden state $\overrightarrow{\boldsymbol{h}_t}$ and backward hidden state $\overleftarrow{\boldsymbol{h}_t}$ as following

$$\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}]. \tag{6.4}$$

The attention mechanism further transforms word representation from $\boldsymbol{h}_t$ to $\boldsymbol{u}_t$ with a one-layer multi-layer perceptron (MLP). The contribution of a word is then rated by a weight α_t evaluated using softmax. Mathematically, we have

$$\boldsymbol{u}_t = \tanh(\boldsymbol{W}_w \boldsymbol{h}_t + \boldsymbol{b}_w), \tag{6.5}$$

$$\alpha_t = \frac{\exp(\boldsymbol{u}_t^\top \boldsymbol{u}_w)}{\sum_t (\exp(\boldsymbol{u}_t^\top \boldsymbol{u}_w)}, \tag{6.6}$$

where $\boldsymbol{W}_w$ is the transformation matrix of MLP and $\boldsymbol{b}_w$ is its bias term. The weight α_t characterizes the similarity of word representation $\boldsymbol{u}_t$ and context vector $\boldsymbol{u}_w$ under softmax function. The context vector can be regarded as a reference object of $\boldsymbol{u}_t$ for cross-modal learning. For example, in attention-based visual question answering (VQA) [81], the context vector can be directly set as text features to calculate the attention weights on image regions. In our case, nevertheless, we do not wish to couple text and image features at this stage because otherwise the learnt features will have to be generated on-the-fly and cannot be indexed offline for retrieval. Instead, the context vector $\boldsymbol{u}_w$ is randomly initialized and updated subsequently during the learning process. Finally, the title representation $\boldsymbol{f}_{\text{title}}$ is generated by aggregation of weighted word representations as following

$$\boldsymbol{f}_{\text{title}} = \sum_t \alpha_t \boldsymbol{h}_t. \tag{6.7}$$

Ingredient encoder. Recipe usually has a section listing out ingredients, their quantities and optionally the corresponding cooking and cutting methods for food preparation. The ingredients include both visible items on dish (e.g., onion, steak) and non-visible items (e.g., oil, salt). The aim of attention is to align the observations on recipe and food image such that ingredients, which are not visible or do no alter the outlook of a dish, will be assigned lower weights. The learning of ingredient representation, $\boldsymbol{f}_{\text{ingredient}}$, is similar to that of title representation. We first obtain the hidden representation of each ingredient (equations 1 to 4), and followed by quantifying the significance of an ingredient with a numerical weight (equations 6.5 to 6.6). The final representation is generated by weighted aggregation as in Equation 6.7.

Instruction encoder. Cooking instructions are composed of sentence with varying-length written in free form. The descriptions are much denser than title and ingredient list for elaborating cooking steps in details. While rich in information, there might not be direct correspondence between a sentence in cooking instruction and dish appearance. For example, the instruction "heat a 10-inch skillet over medium-high heat" has less effect than "lay two slices of bacon over the top" in the final food appearance. The importance should also not be directly impacted by sentence length. For example, the short sentence "bake for 1 hour" could change the dish outlook and should be assigned a higher weight. To this end, the attention mechanism aims to evaluate the relevancy between a sentence and food presentaiton, and meanwhile, the relevancy is also characterized by the importance of words in the sentence. This basically establishes a two-level hierarchy similar to [17] that propagates the contributions of words to sentence level and then sentences to dish appearance for forming recipe representation.

The same procedure as title and ingredient is adopted for word-level representation learning (equations 6.1 to 6.7) to generate sentence vectors, denoted as $\boldsymbol{s}_i$,

$i \in [1, L]$, where L is the number of sentences in the cooking instruction. The sentence-level representations are further aggregated into a vector using a similar procedure. Precisely, the bidirectional forward and backward GRUs followed by one-layer MLP are used to generate hidden representation $\boldsymbol{u}_i$ of $\boldsymbol{s}_i$ as following

$$\overrightarrow{\boldsymbol{h}_i} = \overrightarrow{GRU}(\boldsymbol{s}_i), i \in [1, L], \tag{6.8}$$

$$\overleftarrow{\boldsymbol{h}_i} = \overleftarrow{GRU}(\boldsymbol{s}_i), i \in [1, L], \tag{6.9}$$

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}_i}, \overleftarrow{\boldsymbol{h}_i}], \tag{6.10}$$

$$\boldsymbol{u}_i = \tanh(\boldsymbol{W}_s \boldsymbol{h}_i + \boldsymbol{b}_s). \tag{6.11}$$

Denoting $\boldsymbol{u}_s$ as the sentence-level context vector, the relevancy of a sentence is calculated as

$$\alpha_i = \frac{\exp(\boldsymbol{u}_i^\top \boldsymbol{u}_s)}{\sum_i (\exp(\boldsymbol{u}_i^\top \boldsymbol{u}_s)}, \tag{6.12}$$

where $\boldsymbol{W}_s$ is the transformation matrix of MLP, $\boldsymbol{u}_s$ is the context vector. Similar to Equation 6, $\boldsymbol{u}_s$ is randomly initialized and progressively refined during training. The final representation is obtained through

$$\boldsymbol{f}_{\text{instruction}} = \sum_i \alpha_i \boldsymbol{h}_i. \tag{6.13}$$

Recipe representation. We adopt an early fusion strategy to append the three levels of representations as following

$$\boldsymbol{f}_{\text{recipe}} = [\boldsymbol{f}_{\text{title}}, \boldsymbol{f}_{\text{ingredient}}, \boldsymbol{f}_{\text{instruction}}] \tag{6.14}$$

The dimensions of both $\boldsymbol{f}_{\text{title}}$ and $\boldsymbol{f}_{\text{ingredient}}$ are empirically set as 600. As instruction is dense in description, $\boldsymbol{f}_{\text{instruction}}$ is set as a 1,000 dimensional vector. No normalization is applied when concatenating the three vectors into recipe presentation.

6.1.2 Representation of images

The state-of-the-art deep convolutional network, ResNet-50 [44], is used for image feature extraction. As the network is not pre-trained on food images, we fine-tune ResNet-50 with UMPC Food-101 dataset [10], which contains 75,750 training images of 101 food categories. Different from [2], we do not integrate ResNet-50 with recipe representation for end-to-end feature learning. Instead, pool-5 features of ResNet-50 are extracted. The dimension of $\boldsymbol{f}_{\text{image}}$ is 2,048.

6.1.3 Joint embedding learning

The aim is to transform both recipe and image representations into vectors with equal number of dimensions for similarity comparison. Two projections are learnt through transformation matrices $\boldsymbol{W}_R$ and $\boldsymbol{W}_v$, as following

$$\boldsymbol{\phi}_R = \tanh(\boldsymbol{W}_R \boldsymbol{f}_{\text{recipe}} + \boldsymbol{b}_R), \tag{6.15}$$

$$\boldsymbol{\phi}_v = \tanh(\boldsymbol{W}_v \boldsymbol{f}_{\text{image}} + \boldsymbol{b}_v), \tag{6.16}$$

where $\boldsymbol{\phi}_R$ and $\boldsymbol{\phi}_v$ are respectively the embedding features of recipe and image, and $\boldsymbol{b}_R$ and $\boldsymbol{b}_v$ are their bias terms. The feature dimension is empirically set as 1,024, which is the same with [2]. With this, cosine similarity is employed to evaluate the closeness between two transformed features. The learning goal is to ensure that a query can always score its true positive as higher as possible than negatives and thus the rank loss function with max margin is employed for update of parameters. Since we target for both image-to-recipe and recipe-to-image retrieval, the input of loss function is composed of two triplets: $\langle \boldsymbol{\phi}_v, \boldsymbol{\phi}_R, \boldsymbol{\phi}_{R^-} \rangle$ and $\langle \boldsymbol{\phi}_R, \boldsymbol{\phi}_v, \boldsymbol{\phi}_{v^-} \rangle$. The first element of the triplet is either an image ($\boldsymbol{\phi}_v$) or a recipe ($\boldsymbol{\phi}_R$) query, followed by a true positive and a negative example of a different modality as the second and the third elements. Let the margin be $\delta \in (0, 1)$, the loss function is defined

as

$$\begin{aligned} L = &\max(0, \delta - \cos(\boldsymbol{\phi}_v, \boldsymbol{\phi}_R) + \cos(\boldsymbol{\phi}_v, \boldsymbol{\phi}_{R^-})) \\ &+ \max(0, \delta - \cos(\boldsymbol{\phi}_R, \boldsymbol{\phi}_v) + \cos(\boldsymbol{\phi}_R, \boldsymbol{\phi}_{v^-})), \end{aligned} \tag{6.17}$$

Note that, in addition to the attention mechanism, the technical difference between this work and [2] are in four aspects. First, we do not adopt end-to-end image feature learning as in [2] for saving GPU memory and training time. Second, rank loss is employed. In our empirical study, rank loss is about three times faster in model convergence than the pairwise cosine similarity loss adopted by [2]. The number of epochs required by rank loss is 70, versus 220 epochs as required by cosine similarity loss for training. Third, [2] does not encode title information but instead ultilizes titles as constraint for regularization (see Section 4.5 for details). Finally, skip-thoughts [82] and LSTM are used in [2] to encode cooking instruction without attention modeling.

6.2 Experiment

6.2.1 Dataset

The experiments are conducted on Recipe1M[1], which is one of the largest datasets that contain both recipes and images. The dataset is compiled from dozens of popular cooking websites such as "allrecipes" [2] and "fine cooking" [3]. We use the preprocessed version of the dataset provided by [2], in which 0.4% duplicate recipes and 2% duplicate images have been removed, for empirical studies. The dataset contains 1,029,720 recipes and 887,536 images, with around 70% of data being labeled as training and the remaining being split equally between validation

[1] http://im2recipe.csail.mit.edu/dataset/

[2] https://www.allrecipes.com/

[3] http://www.finecooking.com/

and testing. The average number of ingredients and instructions per recipe are 9.3 and 10.5 respectively. All recipes are written in English and 33% of them are associated with at least one image. We treat a recipe and its associated image as a pair, and generate at most five pairs for recipes having more than one images. We do not use those recipes without images in our experiments.

6.2.2 Experiment setting

Implementation details. Adam optimizer [83] is employed for model training with learning rate set as 10^{-4}. The margin δ in Equation 6.17 is selected as 0.3 by validation and the mini-batch size is set as 128. Per-batch online triplet sampling is employed during training. In each mini-batch, a recipe (image) is restricted to have exactly one ground-truth image (recipe). Furthermore, for each recipe (image), apart from its ground-truth image (recipe), the remaining images (recipes) are used as negatives for model training. The deep model is implemented on tensorflow platform. As end-to-end learning is only performed between recipe representation and joint embedding learning, the model can be trained on a single NVIDIA Tesla K40 GPU.

Evaluation metrics. We use median retrieval rank (MedR) and recall at top K (R@K) as in [2] for performance evaluation. MedR measures the median rank position among where true positives are returned. Therefore, a lower MedR score indicates higher performance. R@K, on the other hand, calculates the fraction of times that a correct recipe is found within the top-K retrieved candidates. R@K provides an intuitive sense of how quickly a true positive can be located by investigating a subset of the retrieved items. Different from MedR, the performance is directly proportional to the score of R@K.

Testing. Same as [2], we report results for subsets of randomly selected recipe-

Table 6.1: Contributions of different encoders and their combinations on 5K dataset.

	im2recipe				recipe2im			
	MedR	R@1	R@5	R@10	MedR	R@1	R@5	R@10
title	58.2	0.044	0.141	0.217	57.6	0.040	0.137	0.215
ingre.	71.0	0.045	0.135	0.202	70.1	0.042	0.133	0.202
inst.	33.9	0.070	0.202	0.294	33.2	0.066	0.201	0.295
title + ingre.	31.9	0.073	0.215	0.310	31.9	0.074	0.211	0.307
title + inst.	26.6	0.082	0.231	0.331	26.8	0.081	0.234	0.334
ingre. + inst.	30.0	0.079	0.223	0.316	29.0	0.075	0.220	0.316
all	20.0	0.104	0.274	0.382	19.1	0.101	0.272	0.382

image pairs from the test set. In a subset, every pair is issued alternately as image or recipe query to retrieve its counterpart, namely the image-to-recipe (im2recipe) or recipe-to-image (recipe2im) retrieval. To evaluate the scalability of retrieval, the subset sizes are respectively set to be 1K, 5K and 10K pairs. The experiments are repeated 10 times for each size of subset and the mean results are reported.

6.2.3 Ablation studies

Table 6.1 lists the contributions of title, ingredient, instruction and their combinations towards performance improvement. On both im2recipe and recipe2im, instruction attains higher performance than title and ingredient alone in large margin. The result clearly verifies the significance of cooking instructions, which embed processing of ingredients with rich procedural actions, in cross-modal retrieval. Title, which is often highlighted with the key ingredient and major cooking method, surprisingly outperforms ingredient. Title and ingredient, nevertheless, appear to be highly complementary, and combination of them leads to improvement close to the performance of instruction alone. Meanwhile, combining instruction with either title or ingredient also results in improvement, and the best performance is

Table 6.2: Performance of attention modeling on 5K dataset. The signs "+" and "-" indicate the results with and without attention modeling respectively.

	im2recipe								recipe2im							
	MedR		R@1		R@5		R@10		MedR		R@1		R@5		R@10	
	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-
title	58.2	61.5	0.044	0.042	0.141	0.139	0.217	0.211	57.6	58.7	0.040	0.039	0.137	0.134	0.215	0.209
ingredient	71.0	73.0	0.045	0.039	0.135	0.123	0.202	0.192	70.1	72.0	0.042	0.039	0.133	0.126	0.202	0.196
instruction	33.9	36.2	0.070	0.068	0.202	0.198	0.298	0.286	33.2	35.1	0.066	0.065	0.201	0.198	0.295	0.290
all	20.0	22.4	0.104	0.099	0.275	0.265	0.382	0.371	19.1	21.7	0.101	0.098	0.272	0.266	0.382	0.372

achieved by concatenating all the three representations.

Figure 6.2 shows two examples explaining the role of instruction on boosting performance. In Figure 3a, title alone already ranks the true positive at top-3 position. Instruction gives high weights to two sentences "bake until meat is done" and "top meatloaf with jar of gravy." As these sentences somewhat describe the interaction between the ingredients and the associated actions (e.g., bake, top), the true positive is ranked at top-1 position. Ingredient, which misses the keyword "meatloaf", only manages to retrieve dishes with beef. The title "new ranch dip" in Figure 3b does not visually describe the content of dish and hence fails to retrieve any sensible images. Instruction encoder, by giving high weights to "refrigerate 1 hour" and "serve with assorted cut-up vegetables", is able to rank true positive at top-1 position. Interestingly, most of ingredients appear in ingredient list are not mentioned in the cooking procedure. Instead, they are described by the sentence "mix all ingredients" which is ranked as the third highest sentence. Browsing the images retrieved by instruction in Figure 3b, most top-ranked images are with the effect of mixing ingredients and being refrigerated.

6.2.4 Effect of attention

We experiment the impact of attention modeling on cross-modal retrieval. Table 6.2 contrasts the performances on 5K datasets. Note that the results without attention are obtained by average sum of words and sentences. As seen in Table

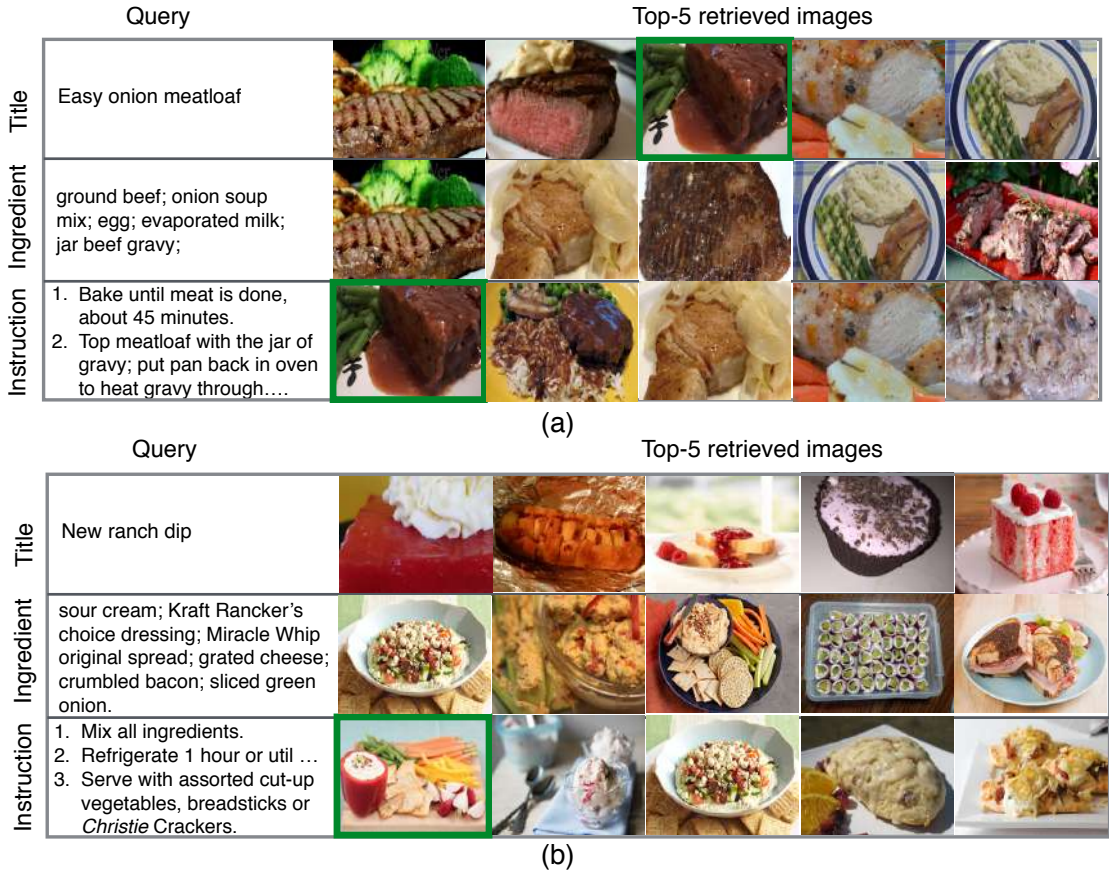


Figure 6.2: Retrieval results by title, ingredient or instruction. True positives are bounded in green box. The highly weighted sentences are listed in the instruction section.

6.2, attention modeling exhibits consistent improvement across different evaluation metrics and levels of comparison. MedR, for example, is averagely upgraded by two ranks for both image-to-recipe and recipe-to-image retrieval. Similar performance is also noted on 1K dataset with MedR being boosted by one position.

Figure 6.3 shows two examples of image-to-recipe retrieval. In the first example, although the word "kalops" in title is assigned lower weight, the true positive is still ranked at top-1 position by attention modeling. This is mainly because sentences 4-7 in the cooking instruction, which characterize the unique way of cooking kalops, are assigned higher weights. Especially, the effects of the operations

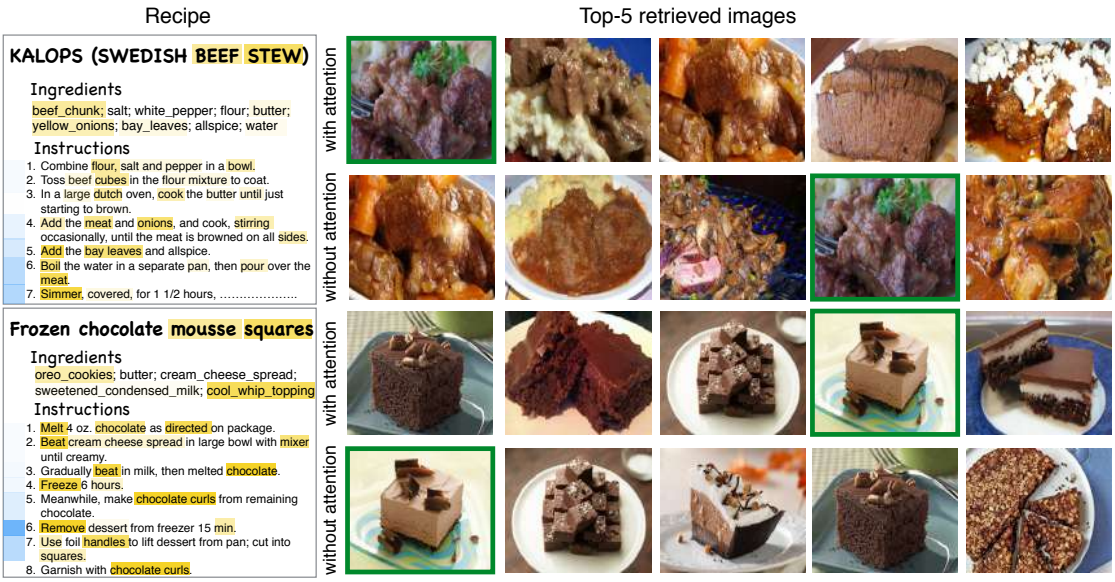


Figure 6.3: Results of recipe-to-image retrieval when attention weights are assigned properly (top) and incorrectly (bottom). The weights of words are highlighted by yellow pen, and the weights of sentences are indicated by blue bar. The intensity of colour indicates the degree of weight.

such as "simmer", "boil water and pour over meat" and "add bay leaves" are partially visible on the dish. Without attention modeling, "pressured cooked beef" will be ranked at top instead. However, when the attention weights are not assigned properly, the result could be worse than without attention modeling as shown in the second example. The keyword "frozen", which characterizes the uniqueness of "mousse square", is not attended in both the title and cooking instruction. Instead, the sentence "remove dessert from freezer" is assigned the highest weight. In this case, although the top-5 retrieved images are all chocolate cakes, the true positive is not ranked at top compared to the method without attention modeling.

6.2.5 Performance comparison

We compare our approach with canonical correlation analysis (CCA) [55], stacked attention network (SAN) [23], joint neural embedding (JNE) [2], and JNE with

semantic regularization (JNE+SR) [2]. We do not compare to classification-based approaches such as the methods presented in Section 3 and Section 5, because only limited number of ingredients, cutting and cooking attributes can be recognized. CCA learns two linear projections for mapping text and image features to a common space that maximizes their feature correlation. The text feature is concatenated from word2vec ingredient vector and skip-thoughts instructor vector provided by [2]. SAN considers ingredient list only and learns the embedding space between ingredient and image features through a two-layer deep attention mechanism. JNE utilizes both ingredients and cooking instructions in joint space learning, but different from our approach, the attention mechanism and title encoder are not considered. JNE+SR is a variant of JNE by imposing a regularization term such that the learnt embedded features will be penalized if failing in performing food categorization. The number of food categories being exploited for SR is 1,047. The categories are semi-automatically compiled from Food-101 dataset [10] and the text mining result on recipe titles of Recipe1M dataset. As the categories are mostly mined from frequent bigrams of titles, we consider that JNE+SR also exploits titles, ingredients and instructions as our approach, except that titles are leveraged in a different stage of learning. We name our approach as attention and also implement attention+SR as a variant based on the 1,047 food categories shared by [2]. Finally, note that different image features are used in these approaches: VGG pool-5 features [42] in SAN, ResNet-50 features [44] fine-tuned by Food-101 dataset, and RestNet-50 features fine-tuned by ImageNet ILSVRC 1000 dataset in JNE.

Table 6.3 lists the detailed performances. Note that we only compare CCA and SAN on 1K dataset. SAN is computationally slow and is not scalable to large dataset. In addition, SAN is designed for image-to-recipe retrieval only. As seen in the results, attention and JNE consistently outperform CCA and SAN across

Table 6.3: Performance comparison of our approach (attention) with various existing methods. The results of JNE and JNE+SR are quoted from [2]. The symbol '-' indicates that the result is not available in [2].

		im2recipe				recipe2im			
		MedR	R@1	R@5	R@10	MedR	R@1	R@5	R@10
1K	random	500	0.001	0.005	0.01	500	0.001	0.005	0.01
	CCA [2]	15.7	0.14	0.32	0.43	24.8	0.09	0.24	0.35
	SAN [23]	16.1	0.125	0.311	0.423	-	-	-	-
	JNE [2]	7.2	0.20	0.45	0.58	6.9	0.20	0.46	0.58
	JNE + SR [2]	5.2	0.24	0.51	0.65	5.1	0.25	0.52	0.65
	attention	4.8	0.254	0.532	0.663	4.7	0.256	0.534	0.667
	attention + SR.	4.6	0.256	**0.537**	**0.669**	**4.6**	**0.257**	**0.539**	**0.671**
5K	JNE [2]	31.5	-	-	-	29.8	-	-	-
	JNE + SR [2]	21.2	-	-	-	20.2	-	-	-
	attention	20.0	0.104	0.274	0.382	19.1	0.101	0.272	0.382
	attention + SR	**19.7**	**0.105**	**0.275**	**0.385**	**19.0**	**0.104**	**0.274**	**0.384**
10K	JNE [2]	62.8	-	-	-	58.8	-	-	-
	JNE + SR [2]	41.9	-	-	-	39.2	-	-	-
	attention	40.7	0.070	0.191	0.274	38.9	0.069	0.192	0.276
	attention + SR	**39.8**	**0.072**	**0.192**	**0.276**	**38.1**	**0.070**	**0.194**	**0.278**

all evaluation metrics on 1K dataset for both im2recipe and recipe2im retrieval. SAN, although adopts attention mechanism, performs considerably worse. This is because SAN considers only ingredients and the image feature learning is based on VGG versus ResNet in other approaches. Our attention approach also outperforms JNE in MedR by raising the median rank for 2 positions, and in R@5 by more than 5.4% of absolute recall improvement. The performance is even slightly better than JNE+SR. When further enhancing our approach with attention+SR, however, only slight improvement is attainable. We speculate that the advantage of SR is limited on our approach because title information has been encoded as attended features for similarity learning. Further imposing food categorization performance, which is equivalent to learning to name food or recipe, in model training can only result in little gain in performance. On the other hand, as no end-to-end learning is conducted between SR and ResNet-50 image features, which could potentially

increases training complexity, the improvement is also expected to be limited. Despite similar performance level as JNE+SR, our deep model is more intuitive than [2] because no ad-hoc compilation of food categorization by semi-automatic text mining is required.

As we move from 1K to 5K and 10K datasets, the performance gap between attention and JNE also gets larger, as indicated in Table 6.3. Our approach with attention manages to boost MedR by 10 and 20 ranks on 5K and 10K datasets, respectively, compared with JNE. When semantic regularization is employed, both approaches improve and attention+SR again outperforms JNE+SR with larger margin as data size increases.

6.2.6 Recipe preprocessing and cross-lingual retrieval

The recipes in Recipe1M dataset are contributed by Internet users and written in free-form. Thus, even extracting ingredient names out of recipes is considered not easy. In the previous experiments, we use the ingredients extracted by bi-directional LSTM as developed in [2] as input to our attention model. With this named-entity extraction technique, for example, *olive_oil* (instead of *olive* or *oil*) will be extracted from the sentence "1 tbsp of olive oil". Nevertheless, the extraction technique sometimes fails to extract ingredients from sentences such as "1 pack udon noodles" or "One 15 oz(240g) can chickpeas, drained and rinsed". Since attention model is capable of assigning weights to words and sentences, we speculate that the effect of noisy texts will be alleviated or even masked out during training. Therefore, instead of explicit preprocessing of recipes, we use raw recipes as input for model learning. In this experiment, we only remove numeric numbers from raw recipes to avoid the explosion of vocabulary size which will adversely affect learning effectiveness. Table 6.4 shows the result that directly

Table 6.4: Results of parsing recipes without (i.e., raw recipe) and with (i.e., preprocessed recipe) named-entity extraction.

		im2recipe				recipe2im			
		MedR	R@1	R@5	R@10	MedR	R@1	R@5	R@10
1K	Raw recipe	4.4	0.259	0.546	0.671	4.2	0.262	0.551	0.677
	Preprocessed recipe	4.8	0.254	0.532	0.663	4.7	0.256	0.534	0.667
5K	Raw recipe	18.1	0.111	0.290	0.402	17.7	0.105	0.293	0.405
	Preprocessed recipe	20.0	0.104	0.274	0.382	19.1	0.101	0.272	0.382
10K	Raw recipe	37.2	0.072	0.202	0.290	35.3	0.069	0.203	0.294
	Preprocessed recipe	40.7	0.070	0.191	0.274	38.9	0.069	0.192	0.276

processing raw recipes can lead to further improvement than using the preprocessed recipes from [2]. The margin of improvement also gets larger with increase of data size. By attention modeling, our approach manages to recover some cases where ingredients are missed by named-entity extraction. In the example of "1 pack udon noodles", "udon" is assigned a relatively higher weight than other words, although our approach is incapable of extracting "udon noodles" as a phrase.

Table 6.5: Cross-lingual retrieval performance.

		MedR	R@1	R@5	R@10
Raw Recipe	Original	4.0	0.273	0.618	0.727
	Translated	8.0	0.218	0.455	0.564
Preprocessed Recipe	Original	4.0	0.291	0.545	0.673
	Translated	14.0	0.109	0.382	0.455

To further test the robustness of attention modeling on noisy text description, we conduct a simulation for cross-lingual recipe retrieval. The simulation is carried out by Google translating the English version recipes into recipes of different languages. We then reverse the process by translating the recipes in different languages back into English version for retrieval. During this process, the text description becomes noisy, for example, "in a large stockpot" becomes "in a big soup pot" and "stir fried bee hoon" becomes "fry fried bees". Table 5 shows the result, where 55 English recipes are subsequently translated from English → Chinese →

Japanese → English and then issued as queries for retrieval on 1K dataset. As expected, the performance of using translated recipes is not as good as the original recipes. When directly processing the raw recipes, the top positives averagely drop by 4 ranks to 8th position in the retrieval list. The result is acceptable because a user can still locate the right recipe within the top-10 retrieved result. Applying named-entity extraction on the translated recipes, on the other hand, suffers larger rank degradation, where the MedR drops from 4th to 14th position. The result basically indicates the resilience of attention modeling in dealing with noisy text description.

6.3 Summary

We have presented a deep hierarchical attention model for understanding of recipes. The model clearly shows the merit of leveraging cooking procedure for retrieval. More importantly, the advantage of attention modeling is evidenced in experiment − higher retrieval performance can be attained when weights are properly assigned to the sentences where their cooking effects are visible on images. Compared with [2], we also show that preprocessing of recipes with named-entity extraction is unnecessary, and indeed, directly processing raw recipes with attention leads to better performance. Currently, our work considers each section of recipes independently, which leads to inconsistency in weight assignment for the same words repeatedly appear in title, ingredient and instruction sections. In addition, co-attention modeling, i.e., assigning weights to both text and image regions, is not explored. Both issues will be the future directions of this work.

CHAPTER 7

CONCLUSION AND FUTURE DIRECTIOONS

In this thesis, we have presented several techniques for the problem of cross model cooking recipe retrieval. This chapter summarizes the main contribution of this thesis. Several promising future directions are also discussed.

7.1 Summary of Contribution

We have contributed to cross-modal cooking recipe retrieval. We address several challenging aspects of the problems. Different from existing techniques that focus on recognizing a pre-defined set of cafeteria or restaurant foods, we focus on recognizing the ingredient as well as the rich attribute of food prepared in the wild, as well as deep understanding of the cooking procedure for cross-modal learning. The major contribution of our works are summarized as follows:

- **Ingredient recognition with multi-task model.** We have proposed deep multi-task models for ingredient recognition in the Chinese food domain. Compared with western food, ingredient recognition in Chinese food domain is much more challenging, as ingredients are usually mixed with each other and present large visual variances due to different cooking and cutting methods. Therefore, we proposed multi-task models for simultaneously learning of ingredient recognition and food categorization. By introducing food category information, the performance of ingredient recognition has been boosted.

- **Zero-shot recipe retrieval with ingredient graph.** We also presented a zero-shot recipe retrieval framework for retrieving recipes from the unknown food category. As our multi-task model is capable of recognizing ingredients, it can be extended to retrieve cooking recipes for foods from unknown categories by ingredient matching. To boost retrieval performance, a graph encoding the contextual relationship among ingredients is learned from the recipe corpus. Using this graph, conditional random field (CRF) is employed to probabilistically tune the probability distribution of ingredients to reduce potential recognition error due to unseen food category. With the aid of external knowledge, the recognized ingredients of a given food picture are matched against a large recipe corpus, for finding appropriate recipes to extract nutrition information. The recipe retrieval performances on unseen food categories demonstrate the feasibility of the proposed approach for zero-shot cooking recipe retrieval.

- **Rich attribute learning.** We have proposed a region-wise multi-scale multi-task learning for rich food attribute (ingredient, cooking and cutting method) recognition. We show that localization of ingredient region is possible even when region-level training examples are not provided. We further leverage all three attributes for cross-modal recipe retrieval and the experimental results validate the merit of rich attributes when comparing to the ingredient-only retrieval techniques.

- **Cross modal learning with stacked attention model.** We also studied the problem of cross-modal retrieval from the viewpoint of cross-modal learning. Specifically, we utilized the stack attention mechanism during the joint space learning. The joint space is learned between attended ingredient regions and ingredients extracted from text recipe and then utilized

for cross-modal retrieval. As the learning occurs at the regional level for image and ingredient level for the recipe, the model has the ability to generalize recipe retrieval to unseen food categories. On an in-house dataset, the proposed model doubled the retrieval performance of DeViSE, a popular cross-modality model but not considering region information during training.

- **Deep understanding of the cooking instructions.** We have proposed a hierarchical attention mechanism for cooking procedure modeling. This is the first attempt in the literature that investigates the extent which attention can deal with the causality effect while being able to demonstrate impressive performance on cross-modal recipe retrieval. In addition, we provided a unified way of dealing with three sections of information (i.e., title, ingredient, instruction) in the recipe.

7.2 Future Directions

As a closure to this thesis, we outline several interesting directions worth to be explored in future research.

- **Leveraging context information for rich attribute learning.** In the thesis, we have studied rich food attribute (i.e., ingredient, cooking and cutting methods) recognition with region-wise multi-task learning model. However, this work considers only region-level identification of ingredients, which results in inconsistent predictions throughout different regions of a dish. One direction to improve the rich attribute prediction is proper utilization of context-level information such as common sense in food preparation. This could be helpful in getting rid of some false predictions.

- **Learning graph based recipe representation.** Cooking recipe contains rich procedure information which can be represented as action/workflow graph [18] [19]. As the action graph defines what actions should be performed on which objects and in what order, it abstracts textual recipes more expressively. Furthermore, any error in cooking procedure can be identified in the early stage when action graph cannot be properly represented or formed a complete workflow. Therefore, a better way to represent recipe is encoding the action graph into a unified representation. Nevertheless, how to encode the action graph into a unified representation for recipe remains a challenge. Fortunately, recent advances in deep learning, such as Graph Convolutional Networks (GCN) [84] [85] proposed for encoding graph, shed light on addressing this problem. Exploiting GCN to learn a unified representation that encodes the action graph of a recipe is a promising direction.

- **Leveraging processed images for multimodal recipe representation learning.** Cooking recipes sometimes provide processed images for each cooking procedure. These processed images are not leveraged, which is one direction worth to be further explored. Intuitively, the processed images are complementary to text cooking instructions. Considering the processed images during the representation learning of cooking recipes will enrich the final recipe representation.

- **Building knowledge graph in the food domain.** Knowledge graph in the food domain, which models the relations among ingredients as well as the interactions among ingredient, cuisine, cooking and cutting methods, has not been explored in this thesis. Nevertheless, we believe that such kind of knowledge graph would be extremely useful for improving the performance of ingredient recognition as well as other food attributes recognition. Also,

having the knowledge graph will facilitate inference of non-visible ingredients.

REFERENCES

[1] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," *arXiv preprint arXiv:1511.02274*, 2015.

[2] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning cross-modal embeddings for cooking recipes and food images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 720, 2017, pp. 619–508.

[3] K. Aizawa, Y. Maruyama, H. Li, and C. Morikawa, "Food balance estimation by using personal dietary tendencies in a multimedia food log," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2176–2185, 2013.

[4] H. Hassannejad, G. Matrella, P. Ciampolini, I. D. Munari, M. Mordonini, and S. Cagnoni, "Automatic diet monitoring: A review of computer vision and wearable sensor-based methods," *International Journal of Food Sciences and Nutrition*, vol. 68, no. 6, pp. 656–670, 2017.

[5] N. Nag, V. Pandey, and R. Jain, "Health multimedia: Lifestyle recommendations based on diverse observations," in *Proceedings of the 2017 ACM International Conference on Multimedia Retrieval*. ACM, 2017, pp. 99–106.

[6] K. Aizawa and M. Ogawa, "Foodlog: Multimedia tool for healthcare applications," *IEEE MultiMedia*, vol. 22, no. 2, pp. 4–8, 2015.

[7] Z.-Y. Ming, J. Chen, Y. Cao, C. Forde, C.-W. Ngo, and T.-S. Chua, "Food photo recognition for dietary tracking: System and experiment," in *Proceedings of the International Conference on Multimedia Modeling*. Springer, 2018, pp. 129–141.

[8] C. K. Martin, T. Nicklas, B. Gunturk, J. B. Correa, H. R. Allen, and C. Champagne, "Measuring food intake with digital photography," *Journal of Human Nutrition and Dietetics*, vol. 27, no. s1, pp. 72–81, 2014.

[9] C. Trattner and D. Elsweiler, "Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 489–498.

[10] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 - mining discriminative components with random forests," in *Proceedings of European Conference on Computer Vision*, 2014, pp. 446–461.

[11] Y. Matsuda and K. Yanai, "Multiple-food recognition considering co-occurrence employing manifold ranking," in *Proceedings of the IEEE International Conference on Pattern Recognition*. IEEE, 2012, pp. 2017–2020.

[12] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "Pfid: Pittsburgh fast-food image dataset," in *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2009, pp. 289–292.

[13] X.-J. Zhang, Y.-F. Lu, and S.-H. Zhang, "Multi-task learning for food identification and analysis with deep convolutional neural networks," *Journal of Computer Science and Technology*, vol. 31, no. 3, pp. 489–500, 2016.

[14] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*. ACM, 2006, pp. 321–330.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 3156–3164.

[16] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[17] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[18] C. Kiddon, G. T. Ponnuraj, L. Zettlemoyer, and Y. Choi, "Mise en place: Unsupervised interpretation of instructional recipes," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 982–992.

[19] Y. Yamakata, S. Imahori, H. Maeta, and S. Mori, "A method for extracting major workflow composed of ingredients, tools, and actions from cooking

procedural text," in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*. IEEE, 2016, pp. 1–6.

[20] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in *Proceedings of the IEEE International Symposium on Multimedia*. IEEE, 2010, pp. 296–301.

[21] J. Chen and C. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in *Proceedings of the 2016 ACM Multimedia Conference*. ACM, 2016, pp. 32–41.

[22] J. Chen, C. Ngo, and T. Chua, "Cross-modal recipe retrieval with rich food attributes," in *Proceedings of the 2017 ACM Multimedia Conference*. ACM, 2017, pp. 1771–1779.

[23] J. Chen, L. Pang, and C.-W. Ngo, "Cross-modal recipe retrieval: How to cook this dish?" in *Proceedings of the International Conference on Multimedia Modeling*. Springer, 2017, pp. 588–600.

[24] J. Chen, L. Pang, and C. Ngo, "Cross-modal recipe retrieval with stacked attention model," *Multimedia Tools and Applications*, pp. 1–17, 2018.

[25] W. Zhang, H. Zhang, T. Yao, Y. Lu, J. Chen, and C. Ngo, "Vireo@ trecvid 2014: Instance search and semantic indexing," in *Proceedings of the NIST TRECVID Workshop*, 2014.

[26] Y. An, Y. Cao, J. Chen, C. Ngo, J. Jia, H. Luan, and T. Chua, "Pic2dish: A customized cooking assistant system," in *Proceedings of the 2017 ACM Multimedia Conference*. ACM, 2017, pp. 1269–1273.

[27] M. Chen, Y. Yang, C. Ho, S. Wang, S. Liu, E. Chang, C. Yeh, and M. Ouhyoung, "Automatic chinese food identification and quantity estimation," in *Proceedings of the SIGGRAPH Asia 2012 Technical Briefs*. ACM, 2012, p. 29.

[28] M. Puri, Z. Zhu, Q. Yu, A. Divakaran, and H. Sawhney, "Recognition and volume estimation of food intake using a mobile device," in *Proceedings of the Workshop on Applications of Computer Vision*. IEEE, 2009, pp. 1–8.

[29] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. Essa, "Leveraging context to support automated food recognition in restaurants," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2015, pp. 580–587.

[30] H. Matsunaga, K. Doman, T. Hirayama, I. Ide, D. Deguchi, and H. Murase, "Tastes and textures estimation of foods based on the analysis of its ingredients list and image," in *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*, 2015, pp. 326–333.

[31] Y. Kawano and K. Yanai, "Real-time mobile food recognition system," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2013.

[32] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International conference on Computer Vision,*, 1999, pp. 1150–1157.

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection." in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[34] M. A. Stricker and M. Orengo, "Similarity of color images," in *Proceedings of the Symposium on Electronic Imaging: Science & Technology*. International Society for Optics and Photonics, 1995, pp. 381–392.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[37] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy, "Im2calories: towards an automated mobile vision food diary," in *Proceedings of IEEE International Conference on Computer Vision*, 2015, pp. 1233–1241.

[38] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops*. IEEE, 2015, pp. 1–6.

[39] X. Wang, D. Kumar, N. Thome, M. Cord, and F. Precioso, "Recipe recognition with large multimodal food dataset," in *Proceedings of International Conference on Multimedia and Expo Workshop*, 2015, pp. 1–6.

[40] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment," in *Proceedings of the International Conference on Smart Homes and Health Telematics*. Springer, 2016, pp. 37–48.

[41] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-slice residual networks for food recognition," *arXiv preprint arXiv:1612.06543*, 2016.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[45] Y. Kawano and K. Yanai, "Food image recognition with deep convolutional features," in *Proceedings of of ACM UbiComp Workshop on Cooking and Eating Activities*, 2014.

[46] T. Ege and K. Yanai, "Simultaneous estimation of food categories and calories with multi-task cnn," in *Proceedings of the International Conference on Machine Vision Applications*. IEEE, 2017, pp. 198–201.

[47] W. Min, S. Jiang, J. Sang, H. Wang, X. Liu, and L. Herranz, "Being a super cook: Joint food attributes and multi-modal content modeling for recipe retrieval and exploration," *IEEE Transactions on Multimedia*, 2017.

[48] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2249–2256.

[49] F. Kong and J. Tan, "Dietcam: Automatic dietary assessment with mobile camera phones," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 147–163, 2012.

[50] F. Zhou and Y. Lin, "Fine-grained image classification by exploring bipartite-graph labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1124–1133.

[51] R. Xu, L. Herranz, S. Jiang, S. Wang, X. Song, and R. Jain, "Geolocalized modeling for dish recognition," *IEEE Transaction on Multimedia*, vol. 17, no. 8, pp. 1187–1199, 2015.

[52] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov *et al.*, "Devise: A deep visual-semantic embedding model," in *Proceedings of Neural Information Processing Systems*, 2013, pp. 2121–2129.

[53] A. Karpathy, A. Joulin, and F. F. Li, "Deep fragment embeddings for bidirectional image sentence mapping," in *Proceedings of Neural Information Processing Systems*, 2014, pp. 1889–1897.

[54] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of ACM International Conference on Multimedia*, 2010, pp. 251–260.

[55] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[56] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," *Subspace, latent structure and feature selection*, pp. 34–51, 2006.

[57] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *International Journal on Computer Vision*, vol. 106, no. 2, pp. 210–233, 2014.

[58] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, "Deep canonical correlation analysis." in *Proceedings of International Conference on Machine Learning*, 2013, pp. 1247–1255.

[59] F. Yan and K. Mikolajczyk, "Deep correlation for matching images and text," in *Proceedings of International Conference on Machine Learning*, 2015, pp. 3441–3450.

[60] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 2014, pp. 7–16.

[61] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proceedings of the 2017 ACM International Conference on Multimedia*. ACM, 2017, pp. 154–162.

[62] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[63] J. Harashima, Y. Someya, and Y. Kikuta, "Cookpad image dataset: An image collection as infrastructure for food research," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 1229–1232.

[64] H. Xie, L. Yu, and Q. Li, "A hybrid semantic item model for recipe search by example," in *Proceedings of IEEE International Symposium on Multimedia*, 2010, pp. 254–259.

[65] H. Su, T. Lin, C. Li, M. Shan, and J. Chang, "Automatic recipe cuisine classification by ingredients," in *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 565–570.

[66] T. Maruyama, Y. Kawano, and K. Yanai, "Real-time mobile recipe recommendation system using food ingredient recognition," in *Proceedings of ACM International Workshop on Interactive Multimedia on Mobile and Portable Devices*, 2012, pp. 27–34.

[67] Y. Yamakata, S. Imahori, H. Maeta, and S. Mori, "A method for extracting major workflow composed of ingredients, tools and actions from cooking procedural text," in *Proceedings of the 8th Workshop on Multimedia for Cooking and Eating Activities*, 2016.

[68] W. Wu and J. Yang, "Fast food recognition from videos of eating for calorie estimation," in *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, 2009, pp. 1210–1213.

[69] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, "Menu-match: Restaurant-specific food logging from images," in *Proceedings of IEEE Workshop on Applications of Computer and Vision*, 2015, pp. 844–851.

[70] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proceedings of Neural Information Processing Systems*, 2013, pp. 935–943.

[71] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.

[72] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[73] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[74] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, 2007.

[75] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.

[76] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[77] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *Proceedings of International Conference on Multimedia and Expo*, 2012.

[78] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition." in *Proceedings of International Conference on Machine Learning*, 2014, pp. 647–655.

[79] T. Mikolov and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Neural Information Processing Systems*, 2013.

[80] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[81] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 451–466.

[82] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Proceedings of the Advances in neural information processing systems*, 2015, pp. 3294–3302.

[83] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[84] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," *arXiv preprint arXiv:1804.01622*, 2018.

[85] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 2014–2023.

LIST OF PUBLICATIONS

Journal Publication

- ***J. J. Chen***, L. Pang, C. W. Ngo, Cross-modal recipe retrieval with stacked attention model, Multimedia Tools and Application (2018):1-17.

Conference Publication

- ***J. J. Chen***, C. W. Ngo, F. L. Feng, and T. S. Chua, Deep Understanding of Cooking Procedure for Cross-modal Recipe Retrieval, ACM Multimedia (ACM MM), Seoul, Korea, October, 2018.

- Z. Y. Ming, ***J. J. Chen***, Y. Cao, C. W. Ngo, and T. S. Chua, Food Photo Recognition for Dietary Tracking: System and Experiment, International Conference on Multimedia Modeling (MMM), Bangkok, Thailand, January, 2018.

- Y. S. An, Y. Cao, ***J. J. Chen***, C. W. Ngo, J. Jia, and H. B. Luan, and T. S. Chua, PIC2DISH: A Customized Cooking Assistant System, ACM Multimedia (ACM MM), Mountain View, CA, USA, October, 2017.

- ***J. J. Chen***, C. W. Ngo, and T. S. Chua, Cross-modal Recipe Retrieval with Rich Food Attributes, ACM Multimedia (ACM MM), Mountain View, CA, USA, October, 2017.

- ***J. J. Chen***, L. Pang, and C. W. Ngo, Cross-modal Recipe Retrieval: How to Cook This Dish?, International Conference on Multimedia Modeling (MMM), Reykjavik, Iceland, January, 2017.

- ***J. J. Chen***, and C. W. Ngo, Deep-based ingredient recognition for cooking recipe retrieval, ACM Multimedia (ACM MM), Amsterdam, The Netherlands, October 2016.

- W. Zhang, H. Zhang, T. Yao, Y. J. Lu, ***J. J. Chen***, and C. W. Ngo, VIREO@TRECVID 2014: Instance Search and Semantic Indexing, NIST TRECVID Workshop (TRECVID'14), Orlando, FL, USA, November, 2014.

APPENDIX A

LIST OF INGREDIENTS

Table A.1: List of ingredient names in Vireo Food-172 dataset

American genseng	Aniseed	Apple chunks	Asparagus
Azuki bean	Baby Chinese cabbage	Bacon	Bamboo shoots chunks
Bamboo shoots tips	Banana slices	Barbecued pork chunks	Barbecued pork slices
Batonnet carrot	Batonnet celtuce	Batonnet cucumber	Batonnet eggplant
Batonnet potato	Batonnet radish	Batonnet tenderloin	Bayleaf
Beaf balls	Bean sprouts	Beans	Beef chunks
Beef slices	Beef tripes	Bitter gourd slices	Black chicken chunks
Black fungus	Black pepper powder	Black rice	Black sesame
Blueberry jam	Boild egg slices	Boild eggs	Boiled chicken slices
Bracken	Bread	Broccoli	Carrot brunoise
Celery brunoise	Chicken brunoise	Cucumber brunoise	Garlic sprout brunoise
Ham brunoise	Lentinus edodes brunoise	Onion brunoise	Bullfrog
Button mushroom	Cabbage	Cashew	Cauliflower
Caviar	Celery	Celery leaves	Celery stalk
Celery stalk slices	Celtuce slices	Cheese	Cherry
Cherry tomato	Cherry tomato slices	Chestnut	Chiba beancurd
Chicken chunks	Chicken Feet	Chicken legs	Chicken Wings
Cabbage chiffonade	Green onion chiffonade	Green vegetables chiffonade	Purple cabbage chiffonade
Chili oil	Chili powder	Chili saurce	Chinese cabbage
Chinese Kale	Chinese mahogany	Chinese Parsleycoriander	Chive pieces
Chives	Chopped chives	Chopped fried bread stick	Chopped ginger
Clams	Coconut cake	Coconut water	Codonopsis pilosula
CoixSeed	Cold steamed rice noodles	Coprinus comatus	Cordyceps sinensis
Corn blocks	Corn kernels	Crab	Crab sticks
Crap roe	Crayfish	Crispbread	Crispy sausage
Crucian	Crushed egg crepe	Crushed garlic	Crushed groundnut kernels
Crushed hot and dry chili	Crushed pepper	Crushed preserved egg	Crushed steamed bread
Crystal sugar	Cucumber slices	Cumin powder	Cured meat chunks
Curry	Double-side fried egg	Dried mushroom	Dried pieces of bean curd
Dried sea shrimp	Duck head	Duck neck	Dumplings
Egg cake	Egg drop	Egg yolk	Eggplant
Eggplant slice	Eggplant sticks	Enoki mushroom	Fermented soya beans
Fermented soybean paste	Fermented vegetables	Fern root noodles	Fish
Fish balls	Fish bean curd	Fish chunks	Fish head
Fish slices	Flatbread	Fresh shrimp	Fried bread stick
Fried dough twist	Fried flour	Fried yuba skin	Garlic bulb
Garlic clove	Garlic leaves	Garlic sprout pieces	Gemelli
Ginger slices	Gluten	Gluten chunks	Glutinous rice
Grape	Green beans	Green soybean	Green soybean with shell
Green vegetables	Groundnut kernels	Ham	Hob blocks of carrot
Hob blocks of cucumber	Hob blocks of eggplant	Hob blocks of potato	Hob blocks of radish
Hot and dry pepper	Hot and dry pepper powder	Hot pickled mustard	Julienned carrot
Julienned cucumber	Julienned ginger	Julienned ham	Julienned radish
Kale borecole	Kelp	Ketchup	Kidney bean
Kiwi	Korean chili sauce	Laver	Lemon
Lemongrass	Lentinus edodes	Lentinus edodes slices	Lettuce
Lilium brownii	Lime	Lime leaves	Longan
Loofah	Lotus root box	Lotus root chunks	Lotus root slices
Lotus seeds	Macaroni	Mango	Mango chunks
Mashed pickled radish	Mashed potatoes	Meat balls	Meat stuffing
Millet	Minced beans	Minced green onion	Minced pickled beans
Minced pickled hot pepper	Minced pork	Mint leaf	Mussels
Mutton chunks	Mutton slices	Noodles	Okra slices
Onion slices	Orange slices	Oyster sauce	Pancakes
Parsley	Pea	Pepper	Pepper slices
Perilla crispa tanaka	Pickled hot pepper	Pickled radish slices	Pickled red peppers
Pickled vegetable	Pine nuts	Pineapple	Pleurotus ostreatus
Poplar Mushroom	Pork chunks	Pork floss	Pork intestines
Pork leg	Pork lungs	Pork paste	Pork slices
Potato slices	Preserved egg chunks	Preserved vegetables	Pumpkin blocks

Quail eggs	Radish slices	Radix astragali	Raisin
Red dates	Ribbonfish	Rice	Rice dumpling
Rice noodle	Rice noodle roll	River snail	Salad dressing
Salted egg	Sausage slices	Scallion pancake	Scallop
Scrambled egg	Sea cucumber	Sea sedge	Seared green onion
Seared pepper	Sesame sauce	Shanghai cabbage	Shelled fresh shrimps
Shredded bamboo shoots	Shredded beef tripe	Shredded celtuce	Shredded chicken
Shredded coconut stuffing	Shredded dried bean curd	Shredded egg crepe	Shredded kelp
Shredded onion	Shredded pepper	Pickled bamboo shoot shreds	Shredded pig ears
Shredded pork	Shredded potato	Shrimp balls	Shrimp eggs
Shumai	Sliced carrot	Sliced double-side fried egg	Sliced fatty beef
Sliced ham	Small crispy rice	Small loaf of steamed bread	Snow peas
Sour sauce	Soy sauce	Soya bean	Soya-bean milk
Soya-bean sprout	Spaghetti	Spareribs chunks	Spareribs
Spiced corned egg	Spinach	Spring rolls	Squid pieces
Squid rings	Starch sheet	Steak	Steamed bread
Steamed Bun	Steamed egg custard	Steamed rice powder	Steamed twisted roll
Stewed Pork	Stinky tofu	Strawberry	Streaky pork chunks
Streaky pork slices	Suckling pig	Sweet and sour sauce	Sweet dumplings
Sweet fermented flour paste	Sweet potato chunks	Sweet potato starch noodles	Sweetened bean paste
Tea-leaves	Tenderloin chunks	Tenderloin slices	Tentacles of Squid
Thick broad-bean sauce	Toast	Tofu chunks	Tomato sclices
Vermicelli	Vinegar	Water	Water mellon
Water spinach	White beech mushroom	White congee	White fungus
White gourd chunks	White onion	White sesame	White yam
Whole black chicken	Whole boiled chicken	Whole chicken	Whole green pepper
Whole preserved egg	Wolfberry	Wonton	Yam chunks
Yam slices	Yellow peaches	Yuba	Zanthoxylum fagara
Zucchini slices			